論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Convex Optimization Techniques for Remote Sensing Data Analysis
著者(和文)	長沼一輝
Author(English)	Kazuki Naganuma
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第12782号, 授与年月日:2024年3月26日, 学位の種別:課程博士, 審査員:小野 峻佑,石井 秀明,小野 功,村田 剛志,横田 理央
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第12782号, Conferred date:2024/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Dissertation

Convex Optimization Techniques for Remote Sensing Data Analysis

Kazuki Naganuma

Graduate Major in Artificial Intelligence
School of Computing
Tokyo Institute of Technology

Supervisor: Shunsuke Ono

December, 2023

Abstract

This dissertation proposes convex optimization techniques for remote sensing data analysis. Remote sensing is a measurement technique to reveal the properties of target materials without getting close to them. Observed remote sensing data often suffer from various severe degradations caused by the conditions of the measurement environments and processes. These severe degradations adversely affect a wide range of tasks that provide information contributing to various applications. In addition, measurement costs are often expensive, making it difficult to obtain large amounts of data. Therefore, it is essential to analyze low-quality and small amounts of data to provide helpful information. A promising approach is to formulate analysis tasks as optimization problems and then solve the problems with optimization algorithms. For the further development of remote sensing data analysis, it is essential to flexibly combine multiple data regularizations and degradation characterizations for the designs of optimization problems and to automatically construct algorithms for the optimization problems, i.e., to establish a framework that can handle a wide range of all target data and tasks. The bottleneck in establishing such a framework is due to parameters included in optimization problems and algorithms. Since these parameters significantly affect the analysis performance and running time, they need to be set appropriately. In terms of parameter settings, remote sensing data analysis should be flexible and reliable to cope with the interdependence of parameters within optimization problems (referred to as regularization parameters) and the difficulty of determining parameters of optimization algorithms (referred to as stepsizes). To this end, we establish novel convex optimization techniques for remote sensing data analysis with the following two approaches. First, to eliminate the interdependence of regularization parameters, we introduce constraint modeling of prior knowledge. The appropriate parameters associated with constraints are invariant with the change of data regularizations and degradation characterizations. In addition, the constraint parameters for noise characterization can be determined from the statistical information of noise. Second, we develop a method that employs the structure of an optimization problem to resolve the difficulty of determining stepsizes. This allows us to set the appropriate stepsizes for any optimization problems. Furthermore, the explicit reflection of the structure of an optimization problem improves the reliability of stepsize determination. In Chapter 3, we design a general framework for handling various types of target images and novel characterization for stripe noise, which is often present in remote sensing. Removing stripe noise, i.e., destriping, from remote sensing images is an essential task in terms of visual quality and subsequent processing. To establish a novel destriping framework, we formulate destriping as a nonsmooth convex optimization problem involving a general form of image regularizations and the flatness constraint, which is a newly-introduced stripe noise characterization. The constraint mathematically models that the intensity of each stripe is constant along one direction, resulting in a strong characterization of stripe noise. To solve the optimization problem, we also develop an efficient algorithm based on the preconditioned primal-dual splitting algorithm (P-PDS). 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 HS

images and IR videos. In Chapter 4, we propose a method for designing stepsizes of P-PDS, an efficient algorithm that solves nonsmooth convex optimization problems, called an Operator norm-based design method of Variable-wise Diagonal Preconditioning (OVDP). First, OVDP constructs diagonal stepsizes using (upper bounds) of the operator norms of the linear operators included in optimization problems, thus explicitly reflecting their structures. Furthermore, since OVDP takes a variable-wise preconditioning approach, it keeps any proximity operator analytically computable. We also prove that our stepsizes satisfy the convergence condition of P-PDS. Finally, we demonstrate the effectiveness and usefulness of OVDP through applications for mixed noise removal of HS images, HS unmixing, and graph signal recovery. In Chapter 5, we introduce the constraint modeling and the problem structure-based stepsize design to establish a novel noise-robust method of HS unmixing, which is the process of decomposing an HS image into material-specific spectra (endmembers) and their spatial distributions (abundance maps). Our method employs, in addition to the two existing regularizations for abundance maps, regularizations for the HS image reconstructed by mixing the estimated abundance maps and endmembers. This strategy makes the unmixing process much more robust in highly noisy scenarios, under the assumption that the abundance maps used to reconstruct the HS image with desirable spatio-spectral structure are also expected to have desirable properties. Furthermore, with constraint modeling including our flatness constraint, our method is designed to accommodate a wider variety of noise including stripe noise, which facilitates regularization parameter settings. To solve the formulated optimization problem, we develop an efficient algorithm based on P-PDS with our stepsizes design method based on the problem structure. Experiments on synthetic and real HS images demonstrate the advantages of our method over existing methods.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we will motivate to use convex optimization for remote sensing data analysis, raise limitations of existing methods, and lead to the general objective of this Ph.D. study. This chapter will be concluded by clarifying the contributions of this study with its outline.

1.1 Background

Remote sensing is a measurement technique to reveal the properties of target materials without getting close to them, in other words, to collect data on materials from places that are difficult for people to enter [59]. With the development of various sensors, the measurement methods used in remote sensing have become diverse. Therefore, attempts have been made to reveal the properties of materials not only through visible information, such as grayscale and RGB images (referred to as natural images in this dissertation) but also through special measurement techniques. We raise some examples:

- hyperspectral (HS) imaging [102, 161], which reveals the spectral information of materials by measuring a wide range of light wavelengths, including invisible spectra, in fine detail;
- thermal infrared (IR) imaging [126, 190], which visualizes the spatial distribution of heat by measuring near-infrared light;
- synthetic aperture (SAR) radar measurement [59, 154], which provides satellite image data without bad weather effects (especially clouds) by obtaining microwave radiation.

Thanks to such helpful information, remote sensing can solve a wide range of problems in academic fields such as geoscience and astronautics, industrial fields such as smart agriculture and smart forestry, and even in other fields such as the rapid provision of disaster information and environmental monitoring [34, 171].

Measurement data observed by remote sensing are often contaminated by various severe degradations (e.g., Gaussian-like random noise, Poisson noise, stripe noise, missing or outliers) due to the conditions of the measurement environments and processes. These severe degradations have a significant negative impact on the analysis that provides information contributing to various applications through tasks such as classification, component estimation, and target detection. In addition, the measurement costs are often expensive, making it difficult to obtain large amounts of data. Therefore, it is essential to establish techniques for analyzing low-quality and small amounts of data to provide helpful information.

There are numerous methods for analyzing remote sensing data; this paper classifies them into two main categories: learning-based and optimization-based approaches. The learning-based approach learns networks to capture data features (e.g., networks proposed in [74, 99, 173]) and then uses them for analysis. In the context of natural image analysis, the learning-based approach has allowed us to develop many effective analysis methods [103, 181, 209]. Because of its effectiveness, the approach has been applied to remote sensing data analysis, such as HS image analysis [37,151,166,213], IR video analysis [38,214], and SAR image analysis [46,211]. The revolutionary aspect of this approach is that it allows the automatic extraction of complicated features inherent in data by learning networks. However, learning effective networks requires a large amount of high-quality training data. As mentioned in the last paragraph, such a large amount of high-quality data is often unavailable in remote sensing data analysis. Therefore, the case where the learning-based approach can be applied is very restricted.

On the other hand, the optimization-based approach has also been studied. In this approach, analysis tasks are formulated as optimization problems, and then analysis results can be obtained by solving the optimization problems using optimization algorithms. The optimization problems can explicitly incorporate prior knowledge, such as degradation models, complex measurement processes, and properties present in a variety of data, allowing us to obtain helpful information from a very small amount of data while appropriately handling severe degradation. Therefore, the optimization-based approach can become a powerful tool for remote sensing data analysis.

1.2 Prior Arts

As mentioned above, optimization-based analysis consists of designing an optimization problem and developing an optimization algorithm. Both have been developed to suit specific target images and analysis tasks as follows.

1.2.1 Optimization Problem Aspects

To reflect prior knowledge in remote sensing data analysis, optimization-based methods design optimization problems using functions where the more desirable the properties of the input data are, the smaller their values are. The functions can be roughly classified into two categories: those that capture the nature of data (called regularizations) and those that characterize degradation (called degradation characterizations). A variety of functions have been studied depending on the data and degradation types.

Regularizations

As an effective regularization, the total variation (TV) regularization has been used in image data analysis because it can capture the spatial piecewise smoothness of image data. Specifically, it is the sum of the absolute values of the differences of spatially adjacent pixels of image data, taking smaller values as adjacent pixel values become more similar. Based on this TV regularization, many regularizations have been proposed to specialize in capturing the specific properties of each remote sensing data. In HS image analysis, HTV [204] and SSTV [8] are known as they capture the spectral nature of HS images. Based on them, more advanced regularizations have been studied, such as HSSTV [177] and l_0-l_1HTV [184]. For video data analysis, video-TV regularizations are used, which calculate not only spatial differences but also temporal differences [33,79,122]. In addition, TV-based regularization has also been proposed for SAR image analysis [19,172].

Another regularization is based on low-rankness. In low-rank regularization, matrices with a low-rank structure are created or found from data, and the nuclear norm, a function

of a good convex relaxation of the rank function, was added to the objective functions of optimization problems. For HS image analysis, to capture the spectral correlation, the authors in [208] have modeled the matrix constructed by rearranging three-dimensional data as a low-rank matrix. Furthermore, the analysis methods for time-series data, such as video and time-series SAR data, have adopted low-rankness regularization for data that do not change much over time [91,129]. In addition to matrices, regularizations that capture tensor low-rankness have been also developed [55,60,84,119,175] because remote sensing data are often higher dimensional than two dimensions (e.g., hyperspectral images, IR video data, and time series SAR data).

Many other types of regularizations have been proposed, such as the discrete cosine transform [96, 200], the wavelet transform [153], the combinations of those regularizations [11,61,75,121,128,185]. In remote sensing data analysis, these regularizations are used appropriately depending on analysis tasks and target data.

Degradation Characterizations

On the other hand, degradation characterization is also essential to improve remote sensing data analysis. To handle degradations, an ℓ_2-norm is often used to add a data fidelity term to objective functions for observed data [8,33,36,39,60,61,79,117,128,185,204]. This function is often used in analysis techniques for dealing with Gaussian-like random noise for data analysis, including natural image analysis. In addition, other characterizations have been proposed to model various types of degradations often contaminating remote sensing data. To model noise with large values distributed sparsely, such as outliers and missing values, an ℓ_1-norm is incorporated in objective functions [8, 61, 128, 185]. In addition, stripe noise, superimposed on remote sensing data with constant brightness values in one spatial direction (see Chapter 3 for details), is modeled based on sparsity [185,208] (ℓ_1-type norms), low-rankness [39] (a nuclear norm), and smoothness [117] (a TV-type norm).

1.2.2 Optimization Algorithms Aspects

As mentioned above, many functions designed based on this wide variety of models are incorporated into optimization problems. Most of them are convex to formulate analysis as convex optimization problems. Convex optimization has good convergence properties and thus provides stable analysis results. However, the convex optimization problems of remote sensing data analysis often have complicated structures and contain non-differentiable functions. This indicates that it is impossible to solve them in closed form or to apply an algorithm that assumes that the objective function is differentiable (gradient descent type algorithm), and thus the optimization is not straightforward.

Proximal splitting optimization algorithms have been used to solve the optimization problem. Specifically, using alternating direction method of multipliers (ADMM)-type algorithms [21,58,65], which iteratively updates variables to obtain a solution to the optimization problem, many existing methods have developed specific algorithms solving their optimization problems of the remote sensing data analysis tasks [8,33,36,39,60,61,75,79, 128,177,185,204]. However, ADMM-type algorithms require operator inversion computations (e.g., the inverses of linear operators, matrix inverses, and the generalized inverses of matrices) within each iteration. Whether the operator inversions can be calculated efficiently depends on their concrete forms. For complicated inversions, there are some cases where imposing strong assumptions on the optimization problems makes the computations efficient [33,75,79,128,177], but in many cases, this strategy is difficult to use. In addition, an alternative proximal splitting optimization algorithm to ADMM, primal-dual splitting (PDS)-type algorithms [32, 51, 148, 182] have attracted attention. PDS-type algorithms efficiently solve non-smooth convex optimization problems because they do not require

complicated inversion computations. Therefore, the algorithms have been increasingly used to solve the optimization problems of remote sensing data analysis [6, 176, 206, 219].

1.3 Key Ingredients of Optimization-Based Analysis and Limitations of Existing Methods

As mentioned in the previous section, optimization-based methods have evolved to adapt to specific target data and analysis tasks. This evolution has enriched the methodology for analysis tasks. However, for the further development of remote sensing data analysis, it is essential to flexibly combine multiple regularizations and degradation characterizations for the designs of optimization problems and to automatically construct algorithms for the optimization problem, i.e., to establish a framework that can handle a wide range of all target data and tasks. Such a framework allows us to easily utilize the knowledge developed in each domain for the other analysis tasks of any target data, in other words, to promote the mutual development of each remote sensing data.

The bottleneck in establishing such a framework is due to parameters included in optimization problems and optimization algorithms. Since these parameters significantly affect the analysis performance and running time, they need to be set appropriately. In terms of this parameter setting, remote sensing data analysis should be flexible and reliable.

Flexibility is essential to accommodate the diversity of analysis conditions. In remote sensing data analysis, it is necessary to acquire various types of information under a variety of environments, including a variety of measurement methods to suit information to be acquired, a wide range of analysis tasks for different applications, and various types and intensities of degradations resulting from different sensors and light amounts. Establishing a flexible method of parameter settings tailored to such diverse conditions can improve the convenience of remote sensing data analysis.

Reliability plays an important role in ensuring the validity of information analyzed from severely degraded data. Remote sensing data analysis must provide valid information to help professional investigations or social problem-solving. This implies that the process of information recovery needs to be explainable. With respect to parameter settings, experimental or empirical determination (e.g., in such a way that data somehow look clean) does not ensure the validity of the set values. Instead, appropriate value setting based on, e.g., statistical properties, degradation models, and the structure of the optimization problem, can significantly improve the reliability.

However, existing optimization-based methods of remote sensing data analysis do not ensure the flexibility and reliability requirements: first, parameters are determined by experimental trial and error; second, once parameters are determined, the same parameters are used even if analysis conditions, such as noise intensity, change. This is due to the following two issues with existing methods.

The parameters within optimization problems (referred to as regularization parameters) are interdependent. Existing studies have reflected prior knowledge into analysis by including data-regularization functions and degradation characterization functions as terms in the objective functions of optimization problems [8, 33, 36, 39, 60, 61, 79, 117, 128, 185, 204]. The more such functions are included (i.e., the more properties to be captured), the more regularization parameters are associated with them. Such parameters can be adjusted only by relative weighting. This implies that the mathematical information behind the degradation is difficult to use for the adjustments. In addition, changing the analysis task or the target data requires the use of different functions. In such cases, even if some functions are not changed, the appropriate values of the parameters associated with them will vary.

The optimization algorithm contains parameters (referred to as stepsizes) that are

difficult to adjust. Existing studies have adopted stepsize-sensitive optimization algorithms [6, 8, 33, 36, 39, 60, 61, 75, 79, 128, 176, 177, 185, 204, 206, 219], that is, the convergence speed of the algorithm varies with the stepsizes, and their appropriate values strongly depend on the structure of a target optimization problem. The optimization problems of remote sensing data analysis have a variety of structures due to the diversity of analysis conditions. Therefore, there are no universal values as stepsizes, and they have been manually adjusted according to a designed optimization problem.

1.4 General Objective of This Study

In this dissertation, we establish novel convex optimization techniques for remote sensing data analysis. The discussion in the previous section brings us to the following research question: *Can we exploit degradation models, statistical information, and problem structures to determine appropriate values of parameters?* To this end, we take the following approaches.

First, to eliminate the interdependence of regularization parameters, we reflect prior knowledge of remote sensing data analysis by modeling it as constraints rather than adding functions to an objective function. Unlike the parameters of the objective function, the parameters associated with the constraints can be determined independently of each other. In other words, appropriate constraint parameters can be used even if the objective functions or other constraint conditions change regarding the diversity of target data and analysis tasks. In addition, the constraint parameters for noise characterization can be determined from the statistical information of noise.

Next, we develop a method that employs the structure of an optimization problem in order to resolve the difficulty of tuning the stepsizes of its optimization algorithm. This allows us to determine the appropriate stepsizes for any optimization problem. In addition, the explicit reflection of the structure of an optimization problem improves the reliability of stepsize determination.

This dissertation is organized as follows. First, Chapter 2 gives mathematical preliminaries for convex optimization and remote sensing data analysis. Next, Chapter 3 and Chapter 4 describe the contents related to simplified parameter setting. The details are as follows.

- Chapter 3: We design a general framework for handling various types of target images and novel characterization for stripe noise, which is often present in remote sensing. 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 from examining and activating 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, we propose a general destriping framework using a newly-introduced stripe noise characterization, named *flatness constraint*, 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 flatness constraint. 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 the preconditioned primal-dual splitting algorithm (P-PDS), which

is one of the PDS-type algorithms, and an existing stepsize design method. 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 HS images and IR videos.

- Chapter 4: We propose a method for designing stepsizes of P-PDS, an efficient algorithm that solves nonsmooth convex optimization problems. To speed up the convergence of P-PDS, a design method has been proposed to automatically determine appropriate values of stepsizes from the problem structure. However, the existing method has two limitations. One is that it directly accesses all elements of matrices representing linear operators involved in a given problem, which is inconvenient for handling linear operators implemented as procedures rather than matrices. The other is that it takes an element-wise preconditioning approach, which turns certain types of proximity operators into analytically intractable forms. To overcome these limitations, we establish an Operator norm-based design method of Variable-wise Diagonal Preconditioning (OVDP). First, OVDP constructs diagonal stepsizes using only (upper bounds) of the operator norms of linear operators, thus eliminating the need for their explicit matrix representations. Furthermore, since OVDP takes a variable-wise preconditioning approach, it keeps any proximity operator analytically computable. We also prove that our stepsizes satisfy the convergence condition of P-PDS. Finally, we demonstrate the effectiveness and usefulness of OVDP through applications to mixed noise removal of HS images, HS unmixing, and graph signal recovery.

Then, in Chapter 5, we take advantage of the above techniques to propose a method of unmixing, which is an essential task of remote sensing data analysis. The details are as follows.

- Chapter 5: We establish a novel noise-robust HS unmixing method using constraint modeling and problem structure-based stepsize design. HS unmixing is the process of decomposing an HS image into material-specific spectra (endmembers) and their spatial distributions (abundance maps). Existing unmixing methods have two limitations concerning noise robustness. First, if the input HS image is highly noisy, even if the balance between sparse and piecewise-smooth regularizations for abundance maps is carefully adjusted, noise may remain in the estimated abundance maps or undesirable artifacts may appear. Second, existing methods do not explicitly account for the effects of stripe noise, which is common in HS measurements, in their formulations, resulting in significant degradation of unmixing performance when such noise is present in the input HS image. To overcome these limitations, we propose a new robust hyperspectral unmixing method based on constrained convex optimization. Our method employs, in addition to the two regularizations for the abundance maps, regularizations for the HS image reconstructed by mixing the estimated abundance maps and endmembers. This strategy makes the unmixing process much more robust in highly noisy scenarios, under the assumption that the abundance maps used to reconstruct the HS image with desirable spatio-spectral structure are also expected to have desirable properties. Furthermore, with constraint modeling including our flatness constraint, our method is designed to accommodate a wider variety of noise including stripe noise, which facilitates parameter settings. To solve the formulated optimization problem, we develop an efficient algorithm based on P-PDS with our stepsizes design method based on the problem structure. Experiments on synthetic and real HS images demonstrate the advantages of our method over existing methods.

Finally, we summarize the results obtained in this dissertation and give an outlook on future research in Chapter 6.

Chapter 2

Preliminaries

In this chapter, we give definitions and properties only for vectors. For matrices, change ℓ_2 norms and vector inner products to Frobenius norms and matrix inner products, respectively.

2.1 Notations

Fundatmentals

We write the sets of real numbers and non-negative real numbers in $\mathbb{R}$ and $\mathbb{R}_+$, respectively.

Vectors

We denote vectors by lowercase bold letters (e.g., $\mathbf{x}$). For a vector $\mathbf{x}$, the i-th element of $\mathbf{x}$ is written in x_i or $[\mathbf{x}]_i$.

Matrices

Matrices are denoted by capitalized boldface letters (e.g., $\mathbf{X}$), and the element at the j-th row and i-th column of matrix $\mathbf{X}$ is denoted by $X_{j,i}$ or $[\mathbf{X}]_{j,i}$. For $\mathbf{X} \in \mathbb{R}^{m\times n}$, a matrix $\mathbf{X}^\top \in \mathbb{R}^{n\times m}$ is called the transpose of $\mathbf{X}$ if it satisfies $[\mathbf{X}^\top]_{i,j} = [\mathbf{X}]_{j,i}$ for any $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,m\}$. We denote a matrix $\mathbf{X} \in \mathbb{R}^{\tilde{m}\times\tilde{n}}$ ($\tilde{m} = \sum_{j=1}^{M} m_j, \tilde{n} = \sum_{i=1}^{N} n_i$) consisting of block matrices $\mathbf{X}_{j,i} \in \mathbb{R}^{m_j\times n_i}$ ($j = 1,\dots,M$ and $i = 1,\dots,N$) by $\mathbf{X} = [\mathbf{X}_{j,i}]_{1\le j\le M, 1\le i\le N}$, i.e.,

$$[\mathbf{X}_{j,i}]_{1\le j\le M,1\le i\le N} := \begin{bmatrix} \mathbf{X}_{1,1} & \mathbf{X}_{1,2} & \cdots & \mathbf{X}_{1,N} \\ \mathbf{X}_{2,1} & \mathbf{X}_{2,2} & \cdots & \mathbf{X}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{M,1} & \mathbf{X}_{M,2} & \cdots & \mathbf{X}_{M,N} \end{bmatrix}. \tag{2.1}$$

We then introduce (block) diagonal matrices. Let $\mathbf{x} \in \mathbb{R}^n$. Then, the matrix that has $\mathbf{x}$ as its diagonal elements is denoted by

$$\mathrm{diag}(\mathbf{x}) = \begin{bmatrix} x_1 & & & 0 \\ & x_2 & & \\ & & \ddots & \\ 0 & & & x_n \end{bmatrix} (\in \mathbb{R}^{n\times n}). \tag{2.2}$$

Similarly, for matrices $\mathbf{X}_1 \in \mathbb{R}^{n_1 \times m_1}, \ldots, \mathbf{X}_n \in \mathbb{R}^{n_n \times m_n}$, the matrix that has $\mathbf{X}_1, \ldots, \mathbf{X}_n$ as its diagonal blocks is written in

$$\mathrm{diag}(\mathbf{X}_1, \ldots, \mathbf{X}_n) = \begin{bmatrix} \mathbf{X}_1 & & & \mathbf{O} \\ & \mathbf{X}_2 & & \\ & & \ddots & \\ \mathbf{O} & & & \mathbf{X}_n \end{bmatrix} (\in \mathbb{R}^{\tilde{m} \times \tilde{n}}), \tag{2.3}$$

where $\tilde{m} = \sum_{j=1}^{M} m_j$ and $\tilde{n} = \sum_{i=1}^{N} n_i$.

Cube Data

In this dissertation, we handle cube data of size $n_1 \times n_2 \times n_3$ (e.g., HS images, abundance maps, and IR videos) in a vector form ($\mathbf{x} \in \mathbb{R}^{n_1 n_2 n_3}$) or matrix forms ($\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 n_3}$ or $\mathbf{X} \in \mathbb{R}^{n_1 n_2 \times n_3}$). In all cases, the (i, j, k)-th element of the cube data is denoted by $[\mathbf{x}]_{i,j,k}$ and $[\mathbf{X}]_{i,j,k}$.

2.2 Selected Elements in Linear Algebra

In this dissertation, we denote linear operators by Fraktur letters (e.g., $\mathfrak{L}$). For a linear operator $\mathfrak{L} : \mathbb{R}^m \to \mathbb{R}^n$ that acts on vectors, a matrix $\mathbf{L} \in \mathbb{R}^{n \times m}$ is called the *representation matrix* of $\mathfrak{L}$ if $\mathbf{L}$ satisfies that $\mathbf{L}\mathbf{x} = \mathfrak{L}(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^m$. We introduce some definitions of linear operators.

Difinition 1 (Inner product)**.** The inner product of two vectors $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$, denoted $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$, is defined as

$$\langle \mathbf{x}_1, \mathbf{x}_2 \rangle := \sum_{i=1}^{n} [\mathbf{x}_1]_i [\mathbf{x}_2]_i. \tag{2.4}$$

Similarly, the inner product of two matrices $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^{m \times n}$ is defined as

$$\langle \mathbf{X}_1, \mathbf{X}_2 \rangle := \sum_{i=1}^{n} \sum_{j=1}^{m} [\mathbf{X}_1]_{i,j} [\mathbf{X}_2]_{i,j}. \tag{2.5}$$

Difinition 2 (Adjoint operator)**.** Let $\mathfrak{L} : \mathbb{R}^m \to \mathbb{R}^n$ be a linear operator. A linear operator $\mathfrak{L}^* : \mathbb{R}^n \to \mathbb{R}^m$ is called the adjoint operator of $\mathfrak{L}$ if it satisfies that $\langle \mathfrak{L}(\mathbf{x}), \mathbf{y} \rangle = \langle \mathbf{x}, \mathfrak{L}^*(\mathbf{y}) \rangle$ for any $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^n$. The representation matrix of $\mathfrak{L}^*$ is identical to $\mathbf{L}^\top$.

We define an operator norm closely related to the convergence of a convex optimization algorithm, which will be introduced later.

Difinition 3 (Norms induced by inner product)**.** For vectors, the norm induced by the inner product defined in Eq. (2.4) is called an ℓ_2-norm $\|\cdot\|_2$, which is defined as for $\mathbf{x} \in \mathbb{R}^n$

$$\|\mathbf{x}\|_2 := \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{2.6}$$

For matrices, the norm induced by the inner product defined in Eq. (2.5) is called a Frobenius norm $\|\cdot\|_F$, which is defined as for $\mathbf{X} \in \mathbb{R}^{m \times n}$

$$\|\mathbf{X}\|_F := \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} X_{i,j}^2}. \tag{2.7}$$

Difinition 4 (Operator norm)**.** For a linear operator $\mathfrak{L}$ that acts on vectors, the operator norm $\|\mathfrak{L}\|_{\mathrm{op}}$ is defined by

$$\|\mathfrak{L}\|_{\mathrm{op}} := \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathfrak{L}(\mathbf{x})\|_2}{\|\mathbf{x}\|_2}. \tag{2.8}$$

For a linear operator $\mathfrak{L}$ that acts on matrices, the operator norm of $\mathfrak{L}$ is defined by

$$\|\mathfrak{L}\|_{\mathrm{op}} := \sup_{\mathbf{X}\neq\mathbf{O}} \frac{\|\mathfrak{L}(\mathbf{X})\|_F}{\|\mathbf{X}\|_F}. \tag{2.9}$$

For a representation matrix $\mathbf{A}$, its operator norm satisfies

$$\|\mathbf{A}\|_{\mathrm{op}} := \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sigma_1(\mathbf{A}), \tag{2.10}$$

where $\sigma_1(\mathbf{A})$ is the maximum singular value of $\mathbf{A}$.

Proposition 2.2.1 (The submultiplicity of an operator norm)**.** *Let $\mathfrak{L}_1 \circ \mathfrak{L}_2$ be the composition of linear operators $\mathfrak{L}_1$ and $\mathfrak{L}_2$. The operator norm of $\mathfrak{L}_1 \circ \mathfrak{L}_2$ satisfies that*

$$\|\mathfrak{L}_1 \circ \mathfrak{L}_2\|_{\mathrm{op}} \leq \|\mathfrak{L}_1\|_{\mathrm{op}} \|\mathfrak{L}_2\|_{\mathrm{op}}. \tag{2.11}$$

This property is called the submultiplicity.

2.3 Selected Elements in Convex Optimization

Proper lower-semicontinuous convex functions play a central role in convex optimization.

Difinition 5 (Proper lower-semicontinuous convex function)**.** A function $f : \mathbb{R}^n \to (-\infty, +\infty]$ is said to be *proper lower-semicontinuous convex* if $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^n | f(\mathbf{x}) < \infty\} \neq \emptyset$, $\mathrm{lev}_\alpha(f) := \{x \in \mathbb{R}^n \mid f(\mathbf{x}) \leq \alpha\}$ is closed for any $\alpha \in \mathbb{R}$ and $f(\lambda\mathbf{x} + (1-\lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{y})$ for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and for any $\lambda \in (0, 1)$, respectively.

Then, we define Fenchel–Rockafellar conjugate functions, which also play an important role in convex optimization.

Difinition 6 (Fenchel–Rockafellar conjugate function)**.** Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a proper lower-semicontinuous convex function. *The Fenchel–Rockafellar conjugate function* of f is defined as

$$f^*(\mathbf{x}) := \max_{\mathbf{y}} \langle \mathbf{x}, \mathbf{y} \rangle - f(\mathbf{y}). \tag{2.12}$$

We introduce several typical examples of proper lower-semicontinuous convex functions.

Example 1 (Norms)**.**

(a) (an ℓ_p norm) $\|\cdot\|_p : \mathbb{R}^n \to \mathbb{R}_+ : \mathbf{x} \mapsto (\sum_i^n |x_i|^p)^{1/p}$, where $p \in [1, +\infty)$.

(b) (an ℓ_1 norm of a matrix) $\|\cdot\|_1 : \mathbb{R}^{n\times m} \to \mathbb{R}_+ : \mathbf{X} \mapsto \sum_{i=1}^n \sum_{j=1}^m |X_{i,j}|$.

(c) (a Frobenius norm) $\|\cdot\|_F : \mathbb{R}^{n\times m} \to \mathbb{R}_+$, which is defined in Eq. (2.7).

(d) (a mixed $\ell_{1,2}$-norm grouped by row) $\|\mathbf{X}\|_{1,2,r} : \mathbb{R}^{n\times m} \to \mathbb{R}_+ : \mathbf{X} \mapsto \sum_i \sqrt{\sum_j X_{i,j}^2}$.

(e) (a mixed $\ell_{1,2}$-norm grouped by column) $\|\mathbf{X}\|_{1,2,c} : \mathbb{R}^{n\times m} \to \mathbb{R}_+ : \mathbf{X} \mapsto \sum_j \sqrt{\sum_i X_{i,j}^2}$.

Next, we define convex sets, which are key tools for this study.

Difinition 7 (Convex set)**.** A set $C \subset \mathbb{R}^n$ is called convex if $\lambda \mathbf{x} + (1-\lambda)\mathbf{y} \in C$ for any $\mathbf{x}, \mathbf{y} \in C$ and for any $\lambda \in (0,1)$.

We raise some useful examples of closed convex sets.

Example 2 (Useful closed convex sets)**.**

(a) (An ℓ_p-ball) $B_{p,\varepsilon}^{\mathbf{c}} := \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{c}\|_p \leq \varepsilon\}$, where $\varepsilon \in \mathbb{R}_+$, $\mathbf{c} \in \mathbb{R}^n$, and $p \in [1, +\infty)$.

(b) (A Frobenius-ball) $B_{F,\varepsilon}^{\mathbf{C}} := \{\mathbf{X} \in \mathbb{R}^{n\times m} \mid \|\mathbf{X} - \mathbf{C}\|_F \leq \varepsilon\}$, where $\varepsilon \in \mathbb{R}_+$, $\mathbf{C} \in \mathbb{R}^{n\times m}$.

(c) (A nonnegative orthant) $\mathbb{R}_+^n$.

(d) (Set consisting of one element) $\{\mathbf{a}\}$, where $\mathbf{a} \in \mathbb{R}^n$.

(e) (The kernel of a linear operator) $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{L}(\mathbf{x}) = \mathbf{0}\}$, where $\mathbf{L} : \mathbb{R}^n \to \mathbb{R}^m$ is a linear operator.

Indicator functions allow us to conveniently handle constrained convex optimization problems.

Example 3. (The indicator function of a closed convex set) Let $C \subset \mathbb{R}^n$ be a nonempty set. The *indicator function* of C is defined by

$$\iota_C : \mathbb{R}^n \to \bar{\mathbb{R}} : \mathbf{x} \mapsto \begin{cases} 0, & \text{if } \mathbf{x} \in C; \\ +\infty, & \text{otherwise.} \end{cases} \tag{2.13}$$

If C is closed and convex, ι_C is a proper lower-semicontinuous convex function.

Difinition 8 (Metric Projection onto a closed convex set)**.** For any nonempty closed convex set $C \subset \mathbb{R}^n$, the *metric projection* onto C is defined by

$$P_C : \mathbb{R}^n \to \mathbb{R}^n : \mathbf{x} \mapsto \operatorname*{argmin}_{\mathbf{y}\in C} \|\mathbf{x} - \mathbf{y}\|_2 \tag{2.14}$$

Example 4 (Metric projection instances)**.**

(a) (An ℓ_2-ball)

$$P_{B_{2,\varepsilon}^{\mathbf{c}}}(\mathbf{x}) := \begin{cases} \mathbf{x}, & \text{if } \mathbf{x} \in B_{2,\varepsilon}^{\mathbf{c}}; \\ \mathbf{c} + \frac{\varepsilon(\mathbf{x}-\mathbf{c})}{\|\mathbf{x}-\mathbf{c}\|_2}, & \text{otherwise.} \end{cases} \tag{2.15}$$

(b) (An ℓ_1-ball) $P_{B_{1,\varepsilon}^{\mathbf{c}}}(\mathbf{x})$ can be calculated by efficient ℓ_1-ball projection techniques [53].

(c) (A Frobenius-ball)

$$P_{B_{F,\varepsilon}^{\mathbf{C}}}(\mathbf{X}) := \begin{cases} \mathbf{X}, & \text{if } \mathbf{X} \in B_{F,\varepsilon}^{\mathbf{C}}; \\ \mathbf{C} + \frac{\varepsilon(\mathbf{X}-\mathbf{C})}{\|\mathbf{X}-\mathbf{C}\|_F}, & \text{otherwise.} \end{cases} \tag{2.16}$$

(d) (A nonnegative orthant)

$$[P_{\mathbb{R}_+^n}(\mathbf{x})]_i = \begin{cases} x_i, & \text{if } x_i \geq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

(e) (Set consisting of one element)

$$P_{\{\mathbf{a}\}}(\mathbf{x}) = \mathbf{a}, \tag{2.18}$$

for any $\mathbf{x} \in \mathbb{R}^n$.

2.4 Proximal Tools and Algorithm

2.4.1 Basic Proximal Tools

Let us start with defining the *proximity operator*, which is a key tool in convex optimization algorithms.

Difinition 9 (Proximity operator)**.** Let $f : \mathbb{R}^N \to (-\infty, +\infty]$ be a proximable proper lower-semicontinuous convex function and $\mathbf{G} \in \mathbb{R}^{N\times N}$ be a symmetric and positive definite matrix. The proximity operator of f relative to the metric induced by $\mathbf{G}$ is defined as

$$\mathrm{prox}_{\mathbf{G},f}(\mathbf{x}) := \underset{\mathbf{y}}{\mathrm{argmin}} \frac{1}{2}\langle \mathbf{x}-\mathbf{y}, \mathbf{G}(\mathbf{x}-\mathbf{y})\rangle + f(\mathbf{y}), \tag{2.19}$$

where $\langle\cdot,\cdot\rangle$ is the Euclidean inner product. If $\mathbf{G}$ is a positive scalar matrix, i.e., $\mathbf{G} = \alpha\mathbf{I}$ ($\alpha > 0$), the proximity operator is identical to the standard proximity operator relative to the metric induced by $\mathbf{I}$:

$$\mathrm{prox}_{\mathbf{G},f}(\mathbf{x}) = \mathrm{prox}_{\mathbf{I},\frac{1}{\alpha}f}(\mathbf{x}) = \underset{\mathbf{y}}{\mathrm{argmin}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|_2^2 + \frac{1}{\alpha}f(\mathbf{y}). \tag{2.20}$$

In this paper, the proximity operator relative to the metric induced by a positive matrix that is not a scalar matrix is called the *skewed proximity operator*. In addition, the proximity operator relative to the metric induced by $\mathbf{I}$ is simply called the *proximity operator* and is denoted as

$$\mathrm{prox}_{\mathbf{I},\frac{1}{\alpha}f}(\mathbf{x}) = \mathrm{prox}_{\frac{1}{\alpha}f}(\mathbf{x}). \tag{2.21}$$

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

We would like to note that the standard proximity operators of some popular convex functions, such as the mixed $\ell_{1,2}$-norm and the indicator functions of norm balls, have analytic solutions but their computations are not completely separable element by element. In such cases, even if $\mathbf{G}$ is diagonal (with different elements), the computation of the skewed proximity operator becomes difficult.

Proposition 2.4.1 (The generalization of Moreau's identity [49, Theorem 3.1 (ii)])**.** *Let $f : \mathbb{R}^n \to (-\infty, +\infty]$ be a proximable proper lower-semicontinuous convex function and f^* be its Fenchel–Rockafellar conjugate function. Then, the following equation holds:*

$$\mathrm{prox}_{\mathbf{G},f^*}(\mathbf{x}) = \mathbf{x} - \mathbf{G}^{-1}\mathrm{prox}_{\mathbf{G}^{-1},f}(\mathbf{G}\mathbf{x}). \tag{2.22}$$

This equation indicates that the (skewed) proximity operator of f^* is (skew) proximable if the (skewed) proximity operator of f is (skew) proximable.

We introduce several examples of proximity operators that are often present in remote sensing analysis.

Example 5 (The proximity operator of an ℓ_1-norm)**.** For any $\mathbf{x} \in \mathbb{R}^n$, the proximity operator of $\|\cdot\|_1$ with an index γ is given by for all $i \in \{1,\dots,n\}$

$$[\mathrm{prox}_{\gamma\|\cdot\|_1}(\mathbf{x})]_i = \mathrm{sign}(x_i)\max\{|x_i| - \gamma, 0\}, \tag{2.23}$$

which is the well-known soft-thresholding operation [56].

The skew proximity operator of an ℓ_1-norm is not proximable in general. If its inducing matrix $\mathbf{G}$ is a diagonal matrix, i.e., $\mathbf{G} = \mathrm{diag}(\mathbf{g})$, we can easily calculate the skew proximity operator as

$$[\mathrm{prox}_{\mathbf{G},\gamma\|\cdot\|_1}(\mathbf{x})]_i = \mathrm{sign}(x_i)\max\{|x_i| - g_i^{-1}\gamma, 0\}. \tag{2.24}$$

Example 6. For any $\mathbf{X} \in \mathbb{R}^{n\times m}$, the proximity operator of $\|\cdot\|_{1,2,r}$ with an index γ is given by for all $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,m\}$

$$[\mathrm{prox}_{\gamma\|\cdot\|_{1,2,r}}(\mathbf{A})]_{i,j} = \max(1 - \frac{\gamma}{\sqrt{\sum_{j=1}^{m} A_{i,j}^2}}, 0)A_{i,j}. \tag{2.25}$$

Example 7. For any $\mathbf{X} \in \mathbb{R}^{n\times m}$, the proximity operator of $\|\cdot\|_{1,2,c}$ with an index γ is given by for all $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,m\}$

$$[\mathrm{prox}_{\gamma\|\cdot\|_{1,2,c}}(\mathbf{A})]_{i,j} = \max(1 - \frac{\gamma}{\sqrt{\sum_{i=1}^{n} A_{i,j}^2}}, 0)A_{i,j}. \tag{2.26}$$

Example 8 (The proximity of an indicator function)**.** For any nonempty closed convex set, the proximity operator of an index $\gamma > 0$ of the indicator function ι_C coincides with the metric projection onto C, i.e.,

$$\mathrm{prox}_{\gamma\iota_C}(\mathbf{x}) = P_C(\mathbf{x}). \tag{2.27}$$

2.4.2 Preconditioned Primal-Dual Splitting Algorithm (P-PDS) [148]

We consider a convex optimization problem of the following form:

$$\min_{\substack{\mathbf{x}_1,\dots,\mathbf{x}_N,\\ \mathbf{y}_1,\dots,\mathbf{y}_M}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) + \sum_{j=1}^{M} g_j(\mathbf{y}_j) \text{ s.t. } \begin{cases} \mathbf{y}_1 = \sum_{i=1}^{N} \mathfrak{L}_{1,i}(\mathbf{x}_i), \\ \vdots, \\ \mathbf{y}_M = \sum_{i=1}^{N} \mathfrak{L}_{M,i}(\mathbf{x}_i), \end{cases} \tag{2.28}$$

where $f_i : \mathbb{R}^{n_i} \to (-\infty, +\infty]$ and $g_j : \mathbb{R}^{m_j} \to (-\infty, +\infty]$ are proximable proper lower-semicontinuous convex functions, and $\mathfrak{L}_{j,i} : \mathbb{R}^{n_i} \to \mathbb{R}^{m_j}$ is a linear operator ($\forall i = 1,\dots,N$ and $\forall j = 1,\dots,M$)[1].

Let $\mathbf{x} = [\mathbf{x}_1^\top,\dots,\mathbf{x}_N^\top]^\top \in \mathbb{R}^{\tilde{n}}$ ($\tilde{n} = \sum_{i=1}^{N} n_i$), $\mathbf{y} = [\mathbf{y}_1^\top,\dots,\mathbf{y}_M^\top]^\top \in \mathbb{R}^{\tilde{m}}$ ($\tilde{m} = \sum_{j=1}^{M} m_j$), $f(\mathbf{x}) = \sum_{i=1}^{N} f_i(\mathbf{x}_i)$, $g(\mathbf{y}) = \sum_{j=1}^{M} g_j(\mathbf{y}_j)$, and

$$\mathfrak{L} := \begin{bmatrix} \mathfrak{L}_{1,1} & \mathfrak{L}_{1,2} & \cdots & \mathfrak{L}_{1,N} \\ \mathfrak{L}_{2,1} & \mathfrak{L}_{2,2} & \cdots & \mathfrak{L}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathfrak{L}_{M,1} & \mathfrak{L}_{M,2} & \cdots & \mathfrak{L}_{M,N} \end{bmatrix}. \tag{2.29}$$

P-PDS [148] computes an optimal solution of Prob. (2.28) by the following iterative procedures:

$$\begin{bmatrix} \mathbf{x}^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_1^{-1},f}(\mathbf{x}^{(t)} - \mathbf{\Gamma}_1\mathfrak{L}^*(\mathbf{y}^{(t)})), \\ \mathbf{y}^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_2^{-1},g^*}(\mathbf{y}^{(t)} + \mathbf{\Gamma}_2\mathfrak{L}(2\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)})), \end{bmatrix} \tag{2.30}$$

where $\mathbf{\Gamma}_1 \in \mathbb{R}^{\tilde{n}\times\tilde{n}}$ and $\mathbf{\Gamma}_2 \in \mathbb{R}^{\tilde{m}\times\tilde{m}}$ are symmetric and positive definite matrices called *stepsizes* or *preconditioners*.

If $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ are block-diagonal matrices, that is, $\mathbf{\Gamma}_1 = \mathrm{diag}(\mathbf{\Gamma}_{1,1},\dots,\mathbf{\Gamma}_{1,N})$ and $\mathbf{\Gamma}_2 = \mathrm{diag}(\mathbf{\Gamma}_{2,1},\dots,\mathbf{\Gamma}_{2,M})$ for matrices $\mathbf{\Gamma}_{1,1},\dots,\mathbf{\Gamma}_{1,N},\mathbf{\Gamma}_{2,1},\dots,\mathbf{\Gamma}_{2,M}$ corresponding to

[1]The optimization problem is written using vectors, but P-PDS can handle optimization problems where objective variables are matrices.

$\mathbf{x}_1, \ldots, \mathbf{x}_N, \mathbf{y}_1, \ldots, \mathbf{y}_M$, the procedures in (2.30) can be rewritten as the following equivalent form:

$$\left\lfloor \begin{array}{l} \mathbf{x}_1^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{1,1}^{-1}, f_1}(\mathbf{x}_1^{(t)} - \mathbf{\Gamma}_{1,1} \sum_{j=1}^{M} \mathfrak{L}_{j,1}^*(\mathbf{y}_j^{(t)})), \\ \vdots \\ \mathbf{x}_N^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{1,N}^{-1}, f_N}(\mathbf{x}_N^{(t)} - \mathbf{\Gamma}_{1,N} \sum_{j=1}^{M} \mathfrak{L}_{j,N}^*(\mathbf{y}_j^{(t)})), \\ \mathbf{y}_1^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{2,1}^{-1}, g_1^*}(\mathbf{y}_1^{(t)} + \mathbf{\Gamma}_{2,1} \sum_{i=1}^{N} \mathfrak{L}_{1,i}(2\mathbf{x}_i^{(t+1)} - \mathbf{x}_i^{(t)})), \\ \vdots \\ \mathbf{y}_M^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{2,M}^{-1}, g_M^*}(\mathbf{y}_M^{(t)} + \mathbf{\Gamma}_{2,M} \sum_{i=1}^{N} \mathfrak{L}_{M,i}(2\mathbf{x}_i^{(t+1)} - \mathbf{x}_i^{(t)})). \end{array} \right. \tag{2.31}$$

Compared with (2.30), the procedures in (2.31) can easily be calculated because it avoids the computations of the skewed proximity operators and linear operators over the entire variables.

Here, we introduce the convergence theorem of P-PDS.

Theorem 2.4.1 (Convergence of P-PDS [148, Theorem 1])**.** *Let* $\mathbf{\Gamma}_1$ *and* $\mathbf{\Gamma}_2$ *be symmetric and positive definite matrices satisfying*

$$\left\| \mathbf{\Gamma}_2^{\frac{1}{2}} \circ \mathfrak{L} \circ \mathbf{\Gamma}_1^{\frac{1}{2}} \right\|_{\mathrm{op}}^2 < 1. \tag{2.32}$$

Then, the sequence $(\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)}, \mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_M^{(t)})$ *generated by* (2.30) *converges to an optimal solution* $(\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*, \mathbf{y}_1^*, \ldots, \mathbf{y}_M^*)$ *of Prob.* (2.28)*.*

For the designs of $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$, see Section **??**.

Chapter 3

Flatness Constraints and General Framework for Remote Sensing Image Destriping

3.1 Introduction

Remote Sensing Images such as hyperspectral (HS) images and infrared (IR) videos offer various applications, including mineral detection, earth observation, agriculture, astronomical imaging, automatic target recognition, and video surveillance [16, 20, 195]. 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 [75, 113, 114]. Stripe noise not only degrades visual quality but also seriously affects subsequent processing, such as HS unmixing [20, 123], HS classification [66, 78, 125, 158], and IR video target recognition [108]. 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 [131, 164, 207]. 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 [37, 38, 100, 166, 167]. 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) [92, 115].

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 [8, 33, 36, 43, 184, 204], low-rankness [41, 60, 201, 208, 218], self-similarity [199], sparse representation [162, 187], and combinations of these regularizations [61, 85].

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 [42, 60, 201, 208, 218], a low-rank-based model [39, 80], and a total variation (TV) model [57, 116, 117]. The first model relies on the fact that stripe noise in observed data is (group) sparsely distributed. The second model characterizes stripe noise as low rankness since stripe noise has a strong low-rank structure [39]. The third model captures the vertical (or horizontal) smoothness of stripe noise using TV regularization.

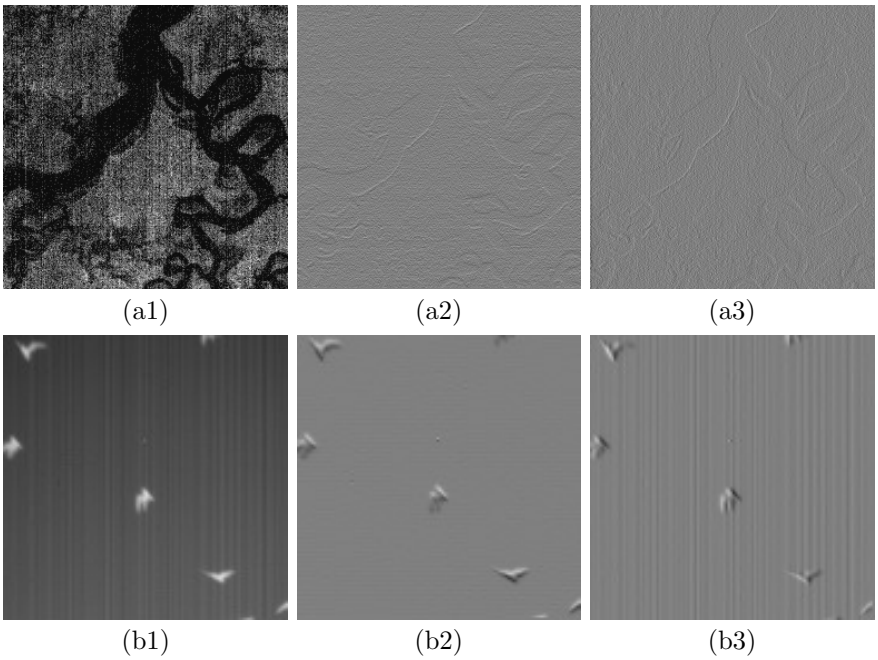


Figure 3.1: Spatial flatness of stripe noise on HS image and IR video data. (a1) Striped HS image data. (a2) Vertical gradient. (a3) Horizontal gradient. (b1) Striped IR video data. (b2) Vertical gradient. (b3) Horizontal gradient.

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 [39], but in the case of destriping methods with the TV model, only the horizontal TV is used to regularize the image [57, 117] 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 hyperspectral image regularization techniques based on spatio-spectral smoothness and correlation [8, 60, 204, 208, 218]. In the case of video data, there are also many regularization techniques that consider moving objects [40, 128, 180]. Combining multiple regularizations is also a promising strategy [61, 121].

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 chapter 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) [148], which can automatically determine the appropriate stepsizes for solving this problem.

The main contributions of the chapter are as follows:

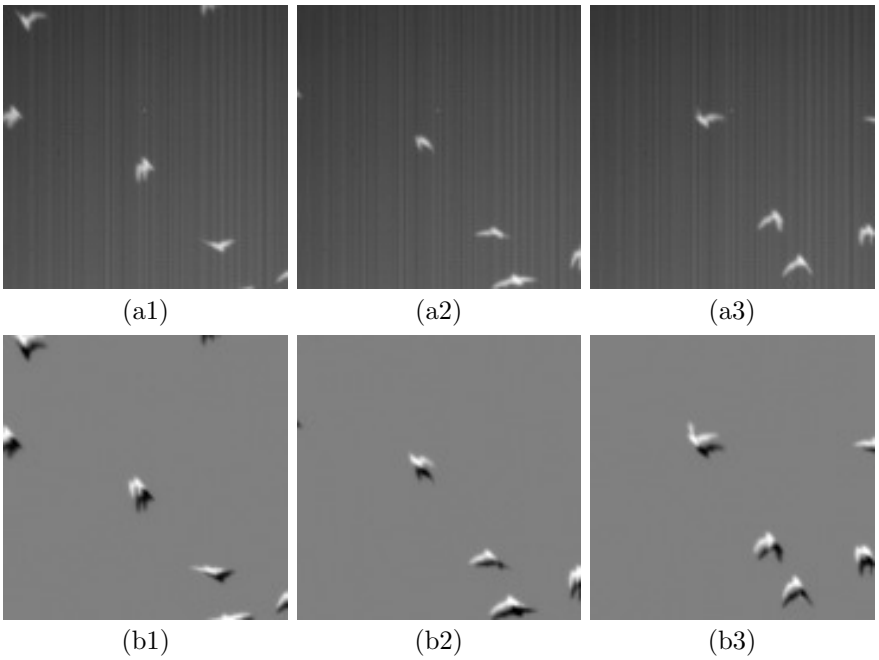


Figure 3.2: Temporal invariance of stripe noise on IR video data. (a1), (a2), and (a3) are frames of a raw data. (b1), (b2), and (b3) are differences in the frames.

- (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.
- (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 HS image, a frame of a raw IR video, and their vertical and horizontal gradients are shown in Fig. 3.1, 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. 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 [27,156]. Some frames of a raw IR video and their differences are shown in Fig. 3.2, 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 flatness constraint along the temporal direction in addition to the spatial constraint. Thanks 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.
- (Automatic stepsize adjustment) Our algorithm can automatically adjust the stepsizes based on the structure of the optimization problem to be solved. In general, the appropriate stepsizes 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 HS images and IR videos.

The remainder of this chapter is organized as follows. Section 3.2 gives reviews the existing sparsity-based, low-rank-based, and TV-based destriping models. Section 3.3

presents the details of the proposed formulation and the solver. Experimental results and discussion are given in Section 3.4. Finally, we summarize the chapter in Section 3.5.

3.2 Review of Existing Approaches

To estimate desirable data from the observed data contaminated by stripe noise and random noise, we model the observation data as follows:

$$\mathbf{v} = \bar{\mathbf{u}} + \bar{\mathbf{l}} + \mathbf{n}, \tag{3.1}$$

where $\bar{\mathbf{u}} \in \mathbb{R}^{n_1 n_2 n_3}$ is a desirable image of interest, $\bar{\mathbf{l}} \in \mathbb{R}^{n_1 n_2 n_3}$ is stripe noise, $\mathbf{n} \in \mathbb{R}^{n_1 n_2 n_3}$ is random noise, and $\mathbf{v} \in \mathbb{R}^{n_1 n_2 n_3}$ is the observed data.

Under the model in (3.1), the destriping problem is often formulated as convex optimization problems with the following form:

$$\min_{\mathbf{u},\mathbf{l}} \sum_{l=1}^{L} R_l(\mathbf{L}_l \mathbf{u}) + \lambda_{\mathbf{l}} J(\mathbf{l}) + \frac{\lambda_{\mathbf{n}}}{2} \|\mathbf{v} - (\mathbf{u} + \mathbf{l})\|_2^2, \tag{3.2}$$

where $R_1(\mathbf{L}_1(\cdot)) : \mathbb{R}^{n_1 n_2 n_3} \to (-\infty, +\infty], \ldots, R_L(\mathbf{L}_L(\cdot)) : \mathbb{R}^{n_1 n_2 n_3} \to (-\infty, +\infty]$ are regularization functions for imaging data with linear operators $\mathbf{L}_1, \ldots, \mathbf{L}_L$ and functions $R_1, \ldots, R_L$, and $J : \mathbb{R}^{n_1 n_2 n_3} \to (-\infty, +\infty]$ is a function characterizing stripe noise, respectively. The positive scalars $\lambda_{\mathbf{l}}$ and $\lambda_{\mathbf{n}}$ 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 [208] is known as a representative work. This method uses the ℓ_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 [85] 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 ℓ_2-norm of each column vector, which groups stripe noise by each column, and thus it is used for the characterization of 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 3.4.3.

The low-rank-based model has been proposed in [39]. In [39], the authors revealed that stripe noise only exists in the horizontal gradient component and that 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 [41, 60, 208, 218].

The TV-based model [57, 117] 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 3.4.3.

3.3 Proposed Framework

The proposed framework involves a general form of regularization term and two types of the flatness constraint. The choice of the specific image regularization and the removal of the temporal flatness constraint are required to fit the nature of an observed image. Depending on image regularization and the temporal flatness constraint, the DP-PDS-based solver needs to be implemented. We illustrate a whole workflow for the proposed framework in Fig. 3.3.

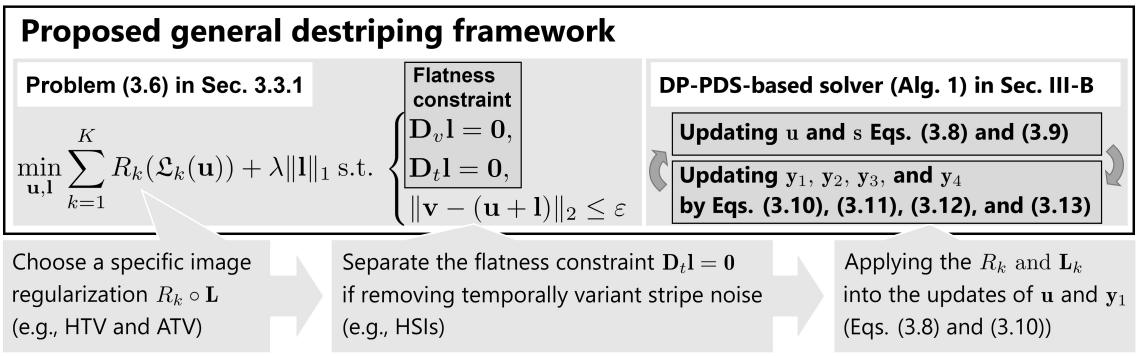


Figure 3.3: A whole workflow of the proposed general destriping framework.

3.3.1 General Destriping Model With Flatness Constraint

In this section, we propose a general destriping model using the flatness constraint. As mentioned, stripe noise $\mathbf{l}$ has the characteristic that the vertical/temporal gradient is zero, i.e.,

$$\begin{cases} \mathbf{D}_v\mathbf{l} = \mathbf{0}, \\ \mathbf{D}_t\mathbf{l} = \mathbf{0}, \end{cases} \tag{3.3}$$

where $\mathbf{D}_v : \mathbb{R}^{n_1 n_2 n_3} \to \mathbb{R}^{(n_1-1)n_2 n_3}$ and $\mathbf{D}_t : \mathbb{R}^{n_1 n_2 n_3} \to \mathbb{R}^{n_1 n_2 (n_3-1)}$ are the vertical and temporal difference operators with the Neumann boundary, which are defined as

$$[\mathbf{D}_v\mathbf{l}]_{i,j,k} := [\mathbf{l}]_{i,j,k} - [\mathbf{l}]_{i+1,j,k}, \begin{cases} \forall i \in \{1, \cdots, n_1 - 1\}, \\ \forall j \in \{1, \cdots, n_2\}, \\ \forall k \in \{1, \cdots, n_3\}, \end{cases} \tag{3.4}$$

$$[\mathbf{D}_t\mathbf{l}]_{i,j,k} := [\mathbf{l}]_{i,j,k} - [\mathbf{l}]_{i,j,k+1}, \begin{cases} \forall i \in \{1, \cdots, n_1\}, \\ \forall j \in \{1, \cdots, n_2\}, \\ \forall k \in \{1, \cdots, n_3 - 1\}. \end{cases} \tag{3.5}$$

Using the flatness constraints in Eq. (3.3), we newly formulate destriping as the following convex optimization problem:

$$\min_{\mathbf{u},\mathbf{l}} \sum_{l=1}^{L} R_l(\mathbf{L}_l\mathbf{u}) + \lambda\|\mathbf{l}\|_1 \text{ s.t. } \begin{cases} \mathbf{D}_v\mathbf{l} = \mathbf{0}, \\ \mathbf{D}_t\mathbf{l} = \mathbf{0}, \\ \mathbf{u} + \mathbf{l} \in B^{\mathbf{v}}_{2,\varepsilon}, \end{cases} \tag{3.6}$$

where $\lambda > 0$ is a hyperparameter, and $R_l(\mathbf{L}_l(\cdot))$ $(\forall l = 1, \cdots, L)$ is a regularization term with a proper semi-continuous convex proximable function R_l and a linear operator $\mathbf{L}_l$. The vertical and temporal gradients of stripe noise are constrained to zero by the first and second constraint, which captures the vertical/temporal flatness of stripe noise. Additionally, we impose the ℓ_1-norm on $\mathbf{l}$ to exploit the sparsity of stripe noise. The third constraint is an ℓ_2 norm constraint with a radius ε for data fidelity to $\mathbf{v}$ given in (3.1). The data-fidelity constraint has an important advantage over the standard additive data fidelity in terms of facilitating hyperparameter settings, as addressed in [7,29,47,138,141]. If stripe noise is variant in the third direction such as HS images, we remove the second constraint.

For data with horizontally featured stripe noise, as in images acquired by whiskbroom scanning [117], we rotate the data 90 degrees in the spatial direction before optimization.

Algorithm 1 The P-PDS algorithm for solving Prob. (3.6)

Input: $\mathbf{v}$, λ, and ε
Output: $\mathbf{u}^{(t)}, \mathbf{l}^{(t)}$

Initialize $\mathbf{u}^{(0)}, \mathbf{l}^{(0)}, \mathbf{y}_1^{(0)}, \ldots, \mathbf{y}_4^{(0)}$;
$t = 0$;
while A stopping criterion is not satisfied **do**
  $\mathbf{u}^{(t+1)} \leftarrow \mathbf{u}^{(t)} - \mathbf{\Gamma}_{\mathbf{u}}(\sum_{l=1}^{L} \mathbf{L}_l^\top \mathbf{y}_{1,l}^{(t)} + \mathbf{y}_4^{(t)})$;
  $\mathbf{l}' \leftarrow \mathbf{l}^{(t)} - \mathbf{\Gamma}_{\mathbf{l}}(\mathbf{D}_v^\top \mathbf{y}_2^{(t)} + \mathbf{D}_t^\top \mathbf{y}_3^{(t)} + \mathbf{y}_4^{(t)})$;
  $\mathbf{l}^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{\mathbf{l}}^{-1}, \lambda\|\cdot\|_1}(\mathbf{l}')$ by Eq. (2.24);
  for $t = 1, \cdots, L$ **do**
    $\mathbf{y}'_{1,l} \leftarrow \mathbf{y}_{1,l}^{(t)} + \mathbf{\Gamma}_{\mathbf{y}_{1,l}} \mathbf{L}_l (2\mathbf{u}^{(t+1)} - \mathbf{u}^{(t)})$;
    $\mathbf{y}_{1,l}^{(t+1)} \leftarrow \mathbf{y}'_{1,l} - \mathbf{\Gamma}_{\mathbf{y}_{1,l}} \mathrm{prox}_{\mathbf{\Gamma}_{\mathbf{y}_{1,l}}, R_l}(\mathbf{\Gamma}_{\mathbf{y}_{1,l}}^{-1} \mathbf{y}'_{1,l})$;
  end for
  $\mathbf{y}_2^{(t+1)} \leftarrow \mathbf{y}_2^{(t)} + \mathbf{\Gamma}_{\mathbf{y}_2} \mathbf{D}_v (2\mathbf{l}^{(t+1)} - \mathbf{l}^{(t)})$;
  $\mathbf{y}_3^{(t+1)} \leftarrow \mathbf{y}_3^{(t)} + \mathbf{\Gamma}_{\mathbf{y}_3} \mathbf{D}_t (2\mathbf{l}^{(t+1)} - \mathbf{l}^{(t)})$;
  $\mathbf{y}_4^{(t)} \leftarrow \mathbf{y}_4^{(t)} + \mathbf{\Gamma}_4 (2(\mathbf{u}^{(t+1)} + \mathbf{l}^{(t+1)}) - (\mathbf{u}^{(t)} + \mathbf{l}^{(t)}))$;
  $\mathbf{y}_4^{(t+1)} \leftarrow \mathbf{y}_4^{(t)} - \mathbf{\Gamma}_4 P_{B_{2,\varepsilon}^{\mathbf{v}}}(\mathbf{\Gamma}_4^{-1} \mathbf{y}_4^{(t)})$ by Eq. (3.14);
  $t \leftarrow t + 1$;
end while

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

In this part, we introduce DP-PDS [148] to solve Prob. (3.6). DP-PDS (see Appendix), which is a diagonally preconditioned version of the primal-dual splitting algorithm [32,51], frees us from tedious stepsize settings. Moreover, the convergence speed of DP-PDS is much faster in general than that of the original PDS algorithm.

To solve Prob. (3.6) with DP-PDS, we rewrite it into the following equivalent problem:

$$\begin{aligned} \min_{\substack{\mathbf{u},\mathbf{l},\mathbf{y}_{1,1},\cdots,\mathbf{y}_{1,L},\\ \mathbf{y}_2,\mathbf{y}_3,\mathbf{y}_4}} \; & \lambda\|\mathbf{l}\|_1 + \sum_{l=1}^{L} R_l(\mathbf{y}_{1,l}) + \iota_{\{\mathbf{0}\}}(\mathbf{y}_2) + \iota_{\{\mathbf{0}\}}(\mathbf{y}_3) + \iota_{B_{2,\varepsilon}^{\mathbf{v}}}(\mathbf{y}_4) \\ \text{s.t.} \; & \begin{cases} \mathbf{y}_{1,1} = \mathbf{L}_1 \mathbf{u}, \\ \vdots \\ \mathbf{y}_{1,L} = \mathbf{L}_L \mathbf{u}, \\ \mathbf{y}_2 = \mathbf{D}_v \mathbf{l}, \\ \mathbf{y}_3 = \mathbf{D}_t \mathbf{l}, \\ \mathbf{y}_4 = \mathbf{u} + \mathbf{l}, \end{cases} \end{aligned} \tag{3.7}$$

where $\iota_{\{\mathbf{0}\}}$ and $\iota_{B_{2,\varepsilon}^{\mathbf{v}}}$ are the indicator functions of $\{\mathbf{0}\}$ and $B_{2,\varepsilon}^{\mathbf{v}}$, respectively. DP-PDS computes the solution of Eq. (3.7) by updating primal variables ($\mathbf{u}$ and $\mathbf{l}$) and dual variables ($\mathbf{y}_{1,1}, \ldots, \mathbf{y}_{1,L}, \mathbf{y}_2, \mathbf{y}_3$, and $\mathbf{y}_4$) alternately.

The primal variables are updated as follows:

$$\mathbf{u}^{(t+1)} \leftarrow \mathbf{u}^{(t)} - \mathbf{\Gamma}_{\mathbf{u}} \left(\sum_{l=1}^{L} \mathbf{L}_l^\top (\mathbf{y}_{1,l}) + \mathbf{y}_4 \right), \tag{3.8}$$

$$\mathbf{l}^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{\mathbf{l}}^{-1}, \lambda\|\cdot\|_1} \left(\mathbf{l}^{(t)} - \mathbf{\Gamma}_{\mathbf{l}} \left(\mathbf{D}_v^\top \mathbf{y}_2^{(t)} + \mathbf{D}_v^\top \mathbf{y}_3^{(t)} + \mathbf{y}_4^{(t)} \right) \right), \tag{3.9}$$

where the constants $\boldsymbol{\Gamma}_{\mathbf{u}}$ and $\boldsymbol{\Gamma}_{\mathbf{l}}$ are stepsize parameters that are called *preconditioners*. The preconditioners are given by the coefficients of the linear operations $\mathbf{L}_1, \ldots, \mathbf{L}_L$ and the difference operators $\mathbf{D}_v, \mathbf{D}_t$ (see Eq. (4.5) in Section **??** for the detailed definitions).

Then, the dual variables are updated as follows:

$$\mathbf{y}_{1,l}^{(t+1)} \leftarrow \mathrm{prox}_{\boldsymbol{\Gamma}_{\mathbf{y}_{1,l}}^{-1}, R_l^*}\left(\mathbf{y}_1^{(t)} + \boldsymbol{\Gamma}_{\mathbf{y}_{1,l}}\left(\mathbf{L}_l\left(2\mathbf{u}^{(t+1)} - \mathbf{u}^{(t)}\right)\right)\right), (\forall l = 1, \cdots, L) \tag{3.10}$$

$$\mathbf{y}_2^{(t+1)} \leftarrow \mathrm{prox}_{\boldsymbol{\Gamma}_{\mathbf{y}_2}^{-1}, \iota_{\{\mathbf{0}\}}^*}\left(\mathbf{y}_2^{(t)} + \boldsymbol{\Gamma}_{\mathbf{y}_2}\left(\mathbf{D}_v\left(2\mathbf{l}^{(t+1)} - \mathbf{l}^{(t)}\right)\right)\right), \tag{3.11}$$

$$\mathbf{y}_3^{(t+1)} \leftarrow \mathrm{prox}_{\boldsymbol{\Gamma}_{\mathbf{y}_3}^{-1}, \iota_{\{\mathbf{0}\}}^*}\left(\mathbf{y}_3^{(t)} + \boldsymbol{\Gamma}_{\mathbf{y}_3}\left(\mathbf{D}_t\left(2\mathbf{l}^{(t+1)} - \mathbf{l}^{(t)}\right)\right)\right), \tag{3.12}$$

$$\mathbf{y}_4^{(t+1)} \leftarrow \mathrm{prox}_{\boldsymbol{\Gamma}_{\mathbf{y}_4}^{-1}, \iota_{B_{2,\varepsilon}^{\mathbf{v}}}^*}\left(\mathbf{y}_4^{(t)} + \boldsymbol{\Gamma}_{\mathbf{y}_4}\left(2\left(\mathbf{u}^{(t+1)} + \mathbf{l}^{(t+1)}\right) - \left(\mathbf{u}^{(t)} + \mathbf{l}^{(t)}\right)\right)\right), \tag{3.13}$$

where the constants $\boldsymbol{\Gamma}_{\mathbf{y}_{1,1}}, \ldots, \boldsymbol{\Gamma}_{\mathbf{y}_{1,L}}, \boldsymbol{\Gamma}_{\mathbf{y}_2}, \boldsymbol{\Gamma}_{\mathbf{y}_3}$, and $\boldsymbol{\Gamma}_{\mathbf{y}_4}$ are preconditioners that can be also determined automatically (see Eq. (4.5) in Section **??**). The skewed proximity operators in Eq. (3.10) are efficiently computed because R_l is a skew proximable function. The skewed proximity operator of $\iota_{\{\mathbf{0}\}}$ in Eqs. (3.11) and (3.12) are calculated as $\mathrm{prox}_{\boldsymbol{\Gamma}, \iota_{\{\mathbf{0}\}}}(\mathbf{x}) = \mathbf{0}$ for any $\mathbf{x} \in \mathbb{R}^{(n_1-1)n_2n_3}$. The skewed proximity operator of $\iota_{B_{2,\varepsilon}^{\mathbf{v}}}$ in Eq. (3.13) is not proximable in general. In our method, all the diagonal entries of the preconditioner $\boldsymbol{\Gamma}_4$ are $\frac{1}{2}$. Hence, the operator $\mathrm{prox}_{\boldsymbol{\Gamma}_4, \iota_{B_{2,\varepsilon}^{\mathbf{v}}}}$ is easily calculated as

$$\begin{aligned}\mathrm{prox}_{\boldsymbol{\Gamma}_4, \iota_{B_{2,\varepsilon}^{\mathbf{v}}}}(\mathbf{x}) &= \mathrm{prox}_{\mathbf{I}, 2\iota_{B_{2,\varepsilon}^{\mathbf{v}}}}(\mathbf{x}) \\ &= P_{B_{2,\varepsilon}^{\mathbf{v}}}(\mathbf{x}) = \begin{cases}\mathbf{x}, & \text{if } \mathbf{x} \in B_{2,\varepsilon}^{\mathbf{v}}, \\ \mathbf{v} + \frac{\varepsilon(\mathbf{x}-\mathbf{v})}{\|\mathbf{x}-\mathbf{v}\|_2}, & \text{otherwise.}\end{cases}\end{aligned} \tag{3.14}$$

Through these update steps, we obtain the solution of Prob. (3.6). We show the detailed algorithms in Algorithm 1. We note that this algorithm can handle a nonconvex optimization problem that contains the proximable nonconvex function such as the ℓ_0-norm and the rank function. However, its convergence, in this case, is not guaranteed.

In temporally variant stripe noise cases, such as an HS image, the temporal constraint is removed. Following the change, the update step in (3.9) will be as follows:

$$\mathbf{l}^{(t+1)} \leftarrow \mathrm{prox}_{\boldsymbol{\Gamma}_{\mathbf{l}}^{-1}, \lambda\|\cdot\|_1}\left(\mathbf{l}^{(t)} - \boldsymbol{\Gamma}_{\mathbf{l}}\left(\mathbf{D}_v^*\mathbf{y}_2^{(t)} + \mathbf{y}_4^{(t)}\right)\right). \tag{3.15}$$

Then, we remove the update step of $\mathbf{y}_3$ (line 9 of Algorithm 1).

3.3.3 Examples of Image Regularizations

We give some examples of image regularization $\sum_{l=1}^{L} R_l(\mathbf{L}_l\mathbf{u})$ in (3.6). First, let us consider HTV [204]. Since the HTV is an image regularization for HS images, we adopt the formulation that does not involve the temporal flatness constraint. The definition of HTV is

$$\|\mathbf{u}\|_{\mathrm{HTV}} := \sum_{i,j}\sqrt{\sum_k [\mathbf{D}_v\mathbf{u}]_{i,j,k}^2 + [\mathbf{D}_h\mathbf{u}]_{i,j,k}^2}. \tag{3.16}$$

Therefore, by letting

$$\mathbf{L}_1 = [\mathbf{D}_v^\top\ \mathbf{D}_h^\top]^\top, \tag{3.17}$$

$$R_1(\mathbf{y}_{1,1}) = \|\mathbf{y}_{1,1}\|_{1,2} = \sum_{i,j}\sqrt{\sum_k [\mathbf{y}_{1,1,1}]_{i,j,k}^2 + [\mathbf{y}_{1,1,2}]_{i,j,k}^2}, \tag{3.18}$$

where $\mathbf{y}_{1,1} = [\mathbf{y}_{1,1,1}^\top \ \mathbf{y}_{1,1,2}^\top]^\top$, we can apply HTV to Prob. (3.6). The update of $\mathbf{u}$ is as follows:

$$\mathbf{u}^{(t+1)} \leftarrow \mathbf{u}^{(t)} - \mathbf{\Gamma}_{\mathbf{u}} \left(\mathbf{D}_v^\top \mathbf{y}_{1,1,1}^{(t)} + \mathbf{D}_h^\top \mathbf{y}_{1,1,2}^{(t)} + \mathbf{y}_4^{(t)} \right), \tag{3.19}$$

Let $[\mathbf{z}_1^\top \ \mathbf{z}_2^\top]^\top = \mathrm{prox}_{\mathbf{\Gamma}_{\mathbf{y}_{1,1}}, \|\cdot\|_{1,2}}(\mathbf{y}_{1,1})$, then each element is calculated as follows: for any $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, and $k \in \{1, \cdots, n_3\}$

$$[\mathbf{z}_1]_{i,j,k} = \max \left\{ 1 - \frac{[\mathbf{g}_{\mathbf{y}_{1,1,1}}]_{i,j,k}}{\sqrt{\sum_{k'} [\mathbf{y}_{1,1,1}]_{i,j,k'}^2 + [\mathbf{y}_{1,1,2}]_{i,j,k'}^2}}, 0 \right\} [\mathbf{y}_{1,1,1}]_{i,j,k}, \tag{3.20}$$

$$[\mathbf{z}_2]_{i,j,k} = \max \left\{ 1 - \frac{[\mathbf{g}_{\mathbf{y}_{1,1,2}}]_{i,j,k}}{\sqrt{\sum_{k'} [\mathbf{y}_{1,1,1}]_{i,j,k'}^2 + [\mathbf{y}_{1,1,2}]_{i,j,k'}^2}}, 0 \right\} [\mathbf{y}_{1,1,2}]_{i,j,k}, \tag{3.21}$$

where $[\mathbf{g}_{\mathbf{y}_{1,1,1}}^\top \ \mathbf{g}_{\mathbf{y}_{1,1,2}}^\top]^\top$ is the diagonal vector of $\mathbf{\Gamma}_{\mathbf{y}_{1,1}}$, i.e., $\mathbf{\Gamma}_{\mathbf{y}_{1,1}} = \mathrm{diag}([\mathbf{g}_{\mathbf{y}_{1,1,1}}^\top \ \mathbf{g}_{\mathbf{y}_{1,1,2}}^\top]^\top)$. Preconditioners are determined as $\mathbf{\Gamma}_{\mathbf{u}} = (\mathrm{diag}(\mathbf{g}_{\mathbf{D}_v^\top}) + \mathrm{diag}(\mathbf{g}_{\mathbf{D}_h^\top}) + \mathbf{I})^{-1}$, $\mathbf{\Gamma}_{\mathbf{l}} = (\mathrm{diag}(\mathbf{g}_{\mathbf{D}_v^\top}) + \mathbf{I})^{-1}$, $\mathbf{\Gamma}_{\mathbf{y}_{1,1}} = 1/2\mathbf{I}$, and $\mathbf{\Gamma}_{\mathbf{y}_2} = 1/2\mathbf{I}$, where

$$[\mathbf{g}_{\mathbf{D}_v^\top}]_{i,j,k} = \begin{cases} 1, & \text{if } i = 1, n_1, \\ 2, & \text{otherwise,} \end{cases} \tag{3.22}$$

$$[\mathbf{g}_{\mathbf{D}_h^\top}]_{i,j,k} = \begin{cases} 1, & \text{if } j = 1, n_2, \\ 2, & \text{otherwise,} \end{cases} \tag{3.23}$$

for any $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, and $k \in \{1, \cdots, n_3\}$. Finally, we obtain a solver for Prob. (3.6) with HTV.

As another example of an IR video case, we consider ATV [33]. ATV is defined as

$$\|\mathbf{u}\|_{\mathrm{ATV}} := \|\mathbf{D}_v \mathbf{u}\|_1 + \|\mathbf{D}_h \mathbf{u}\|_1 + \|\mathbf{D}_t \mathbf{u}\|_1. \tag{3.24}$$

Therefore, we set $\mathbf{L}_1 = \mathbf{D}_v$, $\mathbf{L}_2 = \mathbf{D}_h$, $\mathbf{L}_3 = \mathbf{D}_t$, and $R_1 = R_2 = R_3 = \|\cdot\|_1$ to apply ATV to Eq. (3.7). Then, we update $\mathbf{u}$ as

$$\mathbf{u}^{(t+1)} \leftarrow \mathbf{u}^{(t)} - \mathbf{\Gamma}_{\mathbf{u}} \left(\mathbf{D}_v^\top \mathbf{y}_{1,1}^{(t)} + \mathbf{D}_h^\top \mathbf{y}_{1,2}^{(t)} + \mathbf{D}_t^\top \mathbf{y}_{1,3}^{(t)} + \mathbf{y}_4^{(t)} \right). \tag{3.25}$$

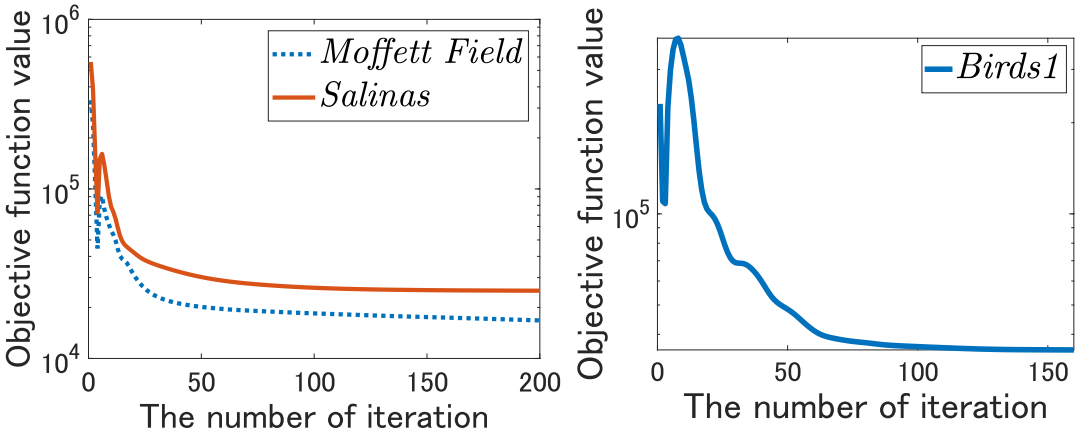
The proximity operator in line 9 of Algorithm 1 is calculated by Eq. (2.24). Preconditioners are set as $\mathbf{\Gamma}_{\mathbf{u}} = (\mathrm{diag}(\mathbf{g}_{\mathbf{D}_v^\top}) + \mathrm{diag}(\mathbf{g}_{\mathbf{D}_h^\top}) + \mathrm{diag}(\mathbf{g}_{\mathbf{D}_t^\top}) + \mathbf{I})^{-1}$, $\mathbf{\Gamma}_{\mathbf{l}} = (\mathrm{diag}(\mathbf{g}_{\mathbf{D}_v^\top}) + \mathrm{diag}(\mathbf{g}_{\mathbf{D}_t^\top}) + \mathbf{I})^{-1}$, $\mathbf{\Gamma}_{\mathbf{y}_{1,1}} = 1/2\mathbf{I}$, $\mathbf{\Gamma}_{\mathbf{y}_{1,2}} = 1/2\mathbf{I}$, $\mathbf{\Gamma}_{\mathbf{y}_{1,3}} = 1/2\mathbf{I}$, $\mathbf{\Gamma}_{\mathbf{y}_2} = 1/2\mathbf{I}$, and $\mathbf{\Gamma}_{\mathbf{y}_3} = 1/2\mathbf{I}$, where $\mathbf{g}_{\mathbf{D}_v^\top}$ and $\mathbf{g}_{\mathbf{D}_t^\top}$ are already defined in the HTV example and

$$[\mathbf{g}_{\mathbf{D}_t^\top}]_{i,j,k} = \begin{cases} 1, & \text{if } k = 1, n_3, \\ 2, & \text{otherwise,} \end{cases} \tag{3.26}$$

for any $i \in \{1, \cdots, n_1\}$, $j \in \{1, \cdots, n_2\}$, and $k \in \{1, \cdots, n_3\}$.

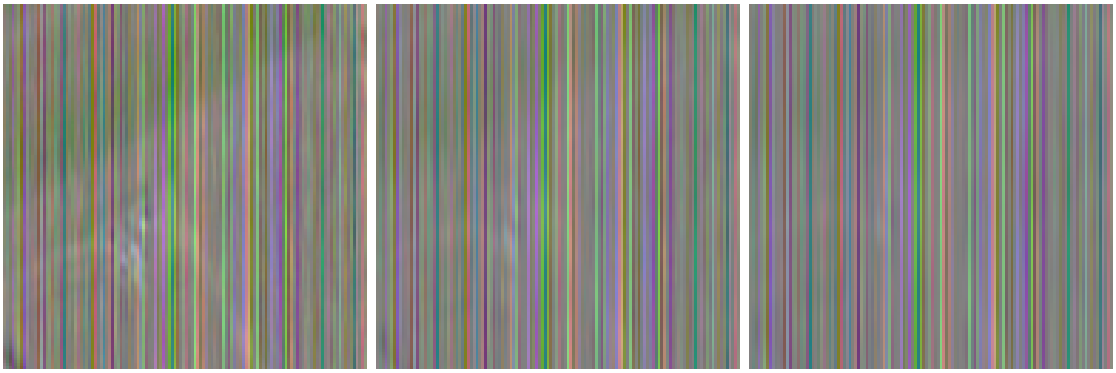
3.3.4 Computational Cost and Running Time

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, 11, 12, 13, and 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 the larger of $O(n_1 n_2 n_3)$ or the one for the image regularization term.



(a) (b)

Figure 3.4: Convergence analysis of DP-PDS for that is experimentally performed using two image regularizations. (a) HSI destriping using HTV (Eq. (5.4)). (b) IR video destriping using ATV (Eq. (3.24)).



(a) Iteration = 20 (b) Iteration = 40 (c) Iteration = 60

Figure 3.5: *Salinas* destriping result of $\mathbf{l}^{(t)}$ in each iteration with HTV (R: 140, G: 101, B: 30).

We measured the actual running times using MATLAB (R2021a) on a Windows 10 computer with an Intel Core i9-10900 3.7GHz processor, 32GB of RAM, and NVIDIA GeForce RTX 3090. The actual running times [s] and total iteration numbers were 13.47 and 932, 5.123 and 317, and 1.064 and 191 for *Moffett Field* destriping using HTV, *Salinas* destriping using HTV, and *Bats1* destriping using ATV, respectively. For the experimental settings, see Sec. 3.4.3.

3.3.5 Convergence Analysis

The convergence of Algorithm 1 is guaranteed by the theorem 2.4.1. Moreover, we experimentally confirm the convergence properties. We plotted the objective function values $\sum_{l=1}^{L} R_l(\mathbf{L}_l\mathbf{u}^{(t)}) + \lambda\|\mathbf{l}^{(t)}\|_1$ versus iterations t on the experiments using HTV and ATV in Fig. 3.4, where our algorithm minimizes the objective function. Figure 3.5 shows *Salinas* destriping results of $\mathbf{l}^{(t)}$ 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.

Table 3.1: All Methods (Stripe Noise Characterization+Image Regularization Examined in Our Experiments)

Image regularization \ Characterization of stripe noise	S [208]	GS [85]	LR [39]	TV [57]	**FC**
HTV (HS image)	[204]	[85]	[39]	[57]	Ours
SSTV (HS image)	[8]	None	None	None	Ours
ASSTV (HS image)	[36]	None	None	None	Ours
TNN (HS image)	[60]	None	None	None	Ours
SSTV+TNN (HS image)	[61]	None	None	None	Ours
l_0-l_1HTV (HS image)	[184]	None	None	None	Ours
ATV (IR video)	[33]	None	None	None	Ours
ITV (IR video)	[33]	None	None	None	Ours
ATV+NN (IR video)	[121]	None	None	None	Ours

3.4 Experiments

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

- Our flatness constraint accurately separates stripe noise from striped images,
- Our framework achieves good destriping performance on average, whatever image regularizations are used.

The specific experimental procedure is as follows.

1. Select image regularizations to be used.
2. Develop DP-PDS-based solvers for all optimization problems that include all combinations of the image regularizations and the stripe noise characterizations summarized in Table 3.1.
3. Set some parameters such as the weight of image regularization, the gradient regularization weight μ of the TV-based model, the data-fidelity parameter ε, and the parameter of the sparse term λ. (Their detailed settings are given in each experimental section).
4. Conduct destriping experiments using these solvers and parameters.

3.4.1 Image Regularizations and Stripe Noise Characterizations

In HS image experiments, we adopted Hyperspectral Total Variation (HTV) [204], Spatio-Spectral Total Variation (SSTV) [8], Anisotropic Spectral-Spatial Total Variation (ASSTV) [36], Tensor Nuclear Norm (TNN) [60], Spatial-Spectral Total Variation with Tensor Nuclear Norm (SSTV+TNN) [61], and l_0-l_1 Hybrid Total Variation (l_0-l_1HTV) [184], which are often used for HS image 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 [61]. In IR video experiments, we adopted Anisotropic Total Variation (ATV), Isotropic Total Variation (ITV) [33], and Anisotropic Total Variation with Nuclear Norm (ATV+NN) [121], which

are known as video regularization. HTV, SSTV, and ASSTV are smoothness-based regularizations, TNN is low-rank-based regularization, SSTV+TNN is the combination of smoothness-based and low-rank-based regularizations, and l_0-l_1HTV is the combination of two smoothness-based regularizations (SSTV [8] and L_0 gradient projection [136]). We compared the proposed flatness constraint (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 short 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 3.1 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).

3.4.2 Dataset Descriptions

We employed three HS image datasets and two IR datasets for experiments in simulated and real noise cases. All images were normalized between $[0, 1]$.

The *Moffett Field* [1] 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 sub-image of size $395 \times 185 \times 176$ (Fig. 3.6 (a)) for experiments in simulated noise cases.

The *Salinas* [3] 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 sub-image of size $360 \times 217 \times 190$ (Fig. 3.8 (a)) for experiments in simulated noise cases.

The *Suwannee* [4] 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 sub-image of size $256 \times 256 \times 360$ (Fig. 3.13 (a)) for experiments in real noise cases.

The *Bats1* and *Bats2* [2], which include hundreds of bats, were collected with three FLIR SC6000 thermal infrared cameras at a frame rate of 125 Hz. For more detailed descriptions, see also [195–197]. We used denoised and raw sub-images of size $256 \times 256 \times 50$ (Figs. 3.7 (a) and 3.13 (b)) for experiments in simulated and real noise cases, respectively.

3.4.3 Experiments in Simulated Noise Cases

For the HS image destriping experiments, the parameter λ of each stripe noise characterization model summarized in Prob. 3.2 was set to a hand-optimized value, so as to achieve the best MPSNR. For a fair comparison, we set ε to the oracle value, i.e., $\varepsilon = \|\mathbf{n}\|_2$. As quantitative evaluations, we employed the mean peak signal-to-noise ratio (MPSNR):

$$\mathrm{MPSNR} = \frac{1}{n_3} \sum_{k=1}^{n_3} 10 \log_{10} \frac{n_1 n_2}{\|\mathbf{u}_k - \bar{\mathbf{u}}_k\|_2^2}, \tag{3.27}$$

and the mean structural similarity overall bands (MSSIM) [188]:

$$\mathrm{MSSIM} = \frac{1}{n_3} \sum_{k=1}^{n_3} \mathrm{SSIM}(\mathbf{u}_k, \bar{\mathbf{u}_k}), \tag{3.28}$$

where $\mathbf{u}_k$ is the kth band of $\mathbf{u}$. The larger these values are, the better the destriping results are. The stopping criterion of Algorithm 1 was set as $\frac{\|\mathbf{u}^{(t+1)}-\mathbf{u}^{(t)}\|_2}{\|\mathbf{u}^{(t)}\|_2} < 1.0 \times 10^{-4}$.

We generated the three types of degraded images:

(i) HS images with vertical stripe noise,

(ii) IR videos with time-invariant vertical stripe noise,

(iii) HS images 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 [164, 207]. For the variety of experiments, we considered the following five types of intensity ranges 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 3.2 and 3.3, 3.4 and 3.5, and 3.6 and 3.7 list the 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.

Figures 3.6, 3.7, and 3.8 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. Figure 3.9 plots their band-wise or frame-wise PSNRs and SSIMs. In the 95th-band results of Figs. 3.9 (a) and 3.9 (b), the 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 Figs. 3.6 (c), 3.6 (d), and 3.6 (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. 3.6 (f)). S-ATV, GS-ATV, LR-ATV, and TV-ATV removed bats as stripe noise, resulting in poor performance (see Figs. 3.7 (c), 3.7 (d), 3.7 (e), and 3.7 (f)). Figures 3.9 (c) and 3.9 (d) show 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. 3.6 (f)) and FC-ATV accurately removed stripe noise, leading to high PSNRs and SSIMs. The SSIM results for Figs. 3.9 (e) and 3.9 (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 (Fig. 3.8 (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.

Figure 3.10 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 between vertical stripe noise and the horizontal lines, leading to worse results. On the other hand, TV and FC obtained better results than LR without capturing the horizontal lines.

Figure 3.11 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.

Figure 3.12 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 flatness constraint, 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, TNN, SSTV+TNN, and l_0-l_1HTV 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.

Table 3.2: MPSNRs of the HS Image Destriping Results in Case (i)

Image data	Range of stripe noise	Regularization function	MPSNR				
			S [208]	GS [85]	LR [39]	TV [57]	**FC**
Salinas	$[-0.2, 0.2]$	HTV	28.70	32.00	**38.29**	37.65	37.35
		SSTV	36.09	36.34	38.61	36.69	**41.00**
		ASSTV	36.92	38.92	**41.64**	37.29	39.29
		TNN	21.78	25.48	16.24	**28.36**	**28.36**
		SSTV+TNN	32.95	34.54	**39.04**	37.58	37.61
		l_0-l_1HTV	36.66	35.96	**41.53**	38.52	39.92
	$[-0.25, 0.25]$	HTV	28.51	31.63	**37.17**	37.21	36.92
		SSTV	35.84	36.10	37.75	36.55	**40.78**
		ASSTV	36.79	38.69	**40.87**	37.18	38.82
		TNN	22.75	24.79	24.86	**28.25**	**28.25**
		SSTV+TNN	32.66	34.26	**37.89**	37.21	37.25
		l_0-l_1HTV	36.38	35.70	**40.87**	38.39	39.69
	$[-0.3, 0.3]$	HTV	28.41	31.46	36.37	**37.07**	36.78
		SSTV	35.73	35.96	36.90	36.39	**40.55**
		ASSTV	36.70	38.62	**40.35**	37.11	38.76
		TNN	22.30	24.06	24.19	**28.04**	**28.04**
		SSTV+TNN	32.53	34.14	37.13	37.13	**37.14**
		l_0-l_1HTV	36.25	35.57	**40.16**	38.24	39.52
	$[-0.35, 0.35]$	HTV	28.34	31.38	36.07	**36.84**	36.62
		SSTV	35.60	35.86	35.86	36.10	**40.07**
		ASSTV	36.52	38.38	**39.68**	36.95	38.54
		TNN	21.83	23.36	23.59	**27.36**	**27.36**
		SSTV+TNN	32.47	34.17	36.71	37.04	**37.14**
		l_0-l_1HTV	36.10	35.45	**39.31**	38.02	39.23
	$[-0.4, 0.4]$	HTV	28.28	31.27	35.02	**36.82**	36.65
		SSTV	35.68	35.96	35.49	36.14	**40.16**
		ASSTV	36.57	38.52	**39.68**	37.02	39.09
		TNN	21.36	22.68	15.51	**26.35**	**26.35**
		SSTV+TNN	32.44	34.14	36.41	37.13	**37.25**
		l_0-l_1HTV	36.26	35.53	38.56	38.30	**39.60**
Moffett Field	$[-0.2, 0.2]$	HTV	27.95	29.32	**36.88**	36.07	36.18
		SSTV	35.33	35.97	38.69	36.17	**40.91**
		ASSTV	30.45	32.68	**44.31**	35.05	38.99
		TNN	24.51	26.26	22.06	**35.54**	**35.54**
		SSTV+TNN	32.63	35.14	**39.94**	37.38	38.00
		l_0-l_1HTV	35.51	35.14	**41.24**	37.77	39.17
	$[-0.25, 0.25]$	HTV	27.62	29.19	**36.36**	36.06	36.19
		SSTV	35.39	36.05	37.86	37.32	**41.07**
		ASSTV	30.37	32.59	**44.07**	33.76	38.97
		TNN	23.9	25.44	31.71	**35.74**	**35.74**
		SSTV+TNN	32.57	35.29	**39.50**	37.34	38.02
		l_0-l_1HTV	35.56	35.24	**40.36**	37.75	39.25
	$[-0.3, 0.3]$	HTV	27.18	29.05	35.72	35.96	**36.07**
		SSTV	35.10	35.79	36.93	35.85	**40.84**
		ASSTV	38.31	39.59	**43.53**	34.81	38.85
		TNN	23.33	24.65	30.66	**36.59**	36.41
		SSTV+TNN	32.33	35.04	**38.72**	37.28	37.95
		l_0-l_1HTV	35.28	34.93	**39.85**	37.69	39.02
	$[-0.35, 0.35]$	HTV	27.20	28.96	35.27	35.83	**35.93**
		SSTV	35.10	35.80	36.42	37.01	**40.47**
		ASSTV	30.27	32.46	**43.10**	33.67	38.78
		TNN	22.80	23.94	30.32	36.73	**36.75**
		SSTV+TNN	32.31	34.95	**38.36**	37.13	37.71
		l_0-l_1HTV	35.34	34.91	**39.33**	37.65	38.99
	$[-0.4, 0.4]$	HTV	26.98	28.80	34.47	35.57	**35.67**
		SSTV	34.96	35.61	35.57	35.61	**40.27**
		ASSTV	30.29	32.47	**42.75**	34.83	38.96
		TNN	22.27	23.26	20.39	35.91	**36.75**
		SSTV+TNN	32.14	34.74	37.24	37.09	**37.66**
		l_0-l_1HTV	35.13	34.74	38.24	37.36	**38.60**

Table 3.3: MSSIMs of the HS Image Destriping Results in Case (i)

Image data	Range of stripe noise	Regularization function	MSSIM				
			S [208]	GS [85]	LR [39]	TV [57]	**FC**
Salinas	$[-0.2, 0.2]$	HTV	0.7601	0.8980	0.9835	**0.9937**	0.9929
		SSTV	0.9344	0.9411	0.9628	0.9266	**0.9751**
		ASSTV	0.9656	0.9756	**0.9922**	0.9646	0.9900
		TNN	0.3230	0.5227	0.4913	**0.9589**	**0.9589**
		SSTV+TNN	0.9285	0.9276	**0.9889**	0.9735	0.9799
		l_0-l_1HTV	0.9515	0.9429	**0.9877**	0.9655	0.9837
	$[-0.25, 0.25]$	HTV	0.8309	0.8949	0.9825	**0.9930**	0.9925
		SSTV	0.9340	0.9407	0.9587	0.9264	**0.9775**
		ASSTV	0.9652	0.9753	**0.9918**	0.9644	0.9791
		TNN	0.4181	0.4739	0.9335	**0.9480**	**0.9480**
		SSTV+TNN	0.9263	0.9260	**0.9852**	0.9737	0.9798
		l_0-l_1HTV	0.9502	0.9412	**0.9873**	0.9651	0.9833
	$[-0.3, 0.3]$	HTV	0.8292	0.8929	0.9817	**0.9928**	0.9924
		SSTV	0.9328	0.9393	0.9509	0.9248	**0.9764**
		ASSTV	0.9648	0.9797	**0.9914**	0.9640	0.9789
		TNN	0.6844	0.7069	**0.9224**	0.9179	0.9179
		SSTV+TNN	0.9252	0.9250	**0.9839**	0.9736	0.9796
		l_0-l_1HTV	0.9492	0.9402	**0.9868**	0.9647	0.9831
	$[-0.35, 0.35]$	HTV	0.8281	0.8916	0.9815	**0.9925**	0.9922
		SSTV	0.9301	0.9374	0.9312	0.9216	**0.9743**
		ASSTV	0.9638	0.9736	**0.9905**	0.9632	0.9781
		TNN	0.3349	0.3871	0.9051	**0.8355**	**0.8355**
		SSTV+TNN	0.9252	0.9257	**0.9824**	0.9727	0.9794
		l_0-l_1HTV	0.9485	0.9393	**0.9858**	0.9641	0.9825
	$[-0.4, 0.4]$	HTV	0.8268	0.8898	0.9796	0.9922	**0.9931**
		SSTV	0.9287	0.9360	0.9353	0.9218	**0.9726**
		ASSTV	0.9638	0.9742	**0.9950**	0.9633	0.9898
		TNN	0.3015	0.3521	0.3331	**0.7340**	0.7339
		SSTV+TNN	0.9238	0.9227	**0.9865**	0.9731	0.9799
		l_0-l_1HTV	0.9488	0.9387	**0.9857**	0.9648	0.9833
Moffett Field	$[-0.2, 0.2]$	HTV	0.6351	0.7237	**0.9199**	0.9165	0.9139
		SSTV	0.8926	0.8952	0.9285	0.8825	**0.9535**
		ASSTV	0.8418	0.8898	**0.9847**	0.9222	0.9691
		TNN	0.4283	0.5301	0.4766	**0.9390**	**0.9390**
		SSTV+TNN	0.8682	0.8857	0.9479	0.9465	**0.9481**
		l_0-l_1HTV	0.8984	0.8834	**0.9551**	0.9269	0.9429
	$[-0.25, 0.25]$	HTV	0.6216	0.7201	0.9151	**0.9165**	0.9144
		SSTV	0.8960	0.8998	0.9218	0.9092	**0.9580**
		ASSTV	0.8411	0.8886	**0.9840**	0.9111	0.9703
		TNN	0.3846	0.4781	0.7496	**0.9342**	**0.9342**
		SSTV+TNN	0.8701	0.8909	0.9450	0.9477	**0.9498**
		l_0-l_1HTV	0.9025	0.8878	**0.9526**	0.9303	0.9466
	$[-0.3, 0.3]$	HTV	0.6100	0.7135	0.9014	0.9133	**0.9107**
		SSTV	0.8871	0.8919	0.9005	0.8711	**0.9548**
		ASSTV	0.9672	0.9689	**0.9819**	0.9201	0.9691
		TNN	0.3546	0.4308	0.7108	**0.8873**	0.8786
		SSTV+TNN	0.8634	0.8851	0.9342	0.9464	**0.9487**
		l_0-l_1HTV	0.8936	0.8795	**0.9410**	0.9243	0.9399
	$[-0.35, 0.35]$	HTV	0.6100	0.7135	0.9014	**0.9133**	0.9107
		SSTV	0.8871	0.8919	0.9005	0.9045	**0.9548**
		ASSTV	0.8399	0.8875	**0.9819**	0.9087	0.9691
		TNN	0.3450	0.4308	0.7108	0.8873	**0.8786**
		SSTV+TNN	0.8634	0.8851	0.9342	0.9464	**0.9487**
		l_0-l_1HTV	0.8972	0.8801	**0.9421**	0.9268	0.9420
	$[-0.4, 0.4]$	HTV	0.5959	0.7015	0.8820	**0.9039**	0.9014
		SSTV	0.8816	0.8871	0.8789	0.8634	**0.9498**
		ASSTV	0.8408	0.8878	**0.9763**	0.9187	0.9691
		TNN	0.2867	0.3602	0.3424	0.8848	**0.8935**
		SSTV+TNN	0.8563	0.8801	0.9179	0.9415	**0.9425**
		l_0-l_1HTV	0.8886	0.8749	0.9254	0.9169	**0.9321**

Table 3.4: MPSNRs of the IR Destriping Results in Case (ii)

IR video data	Range of stripe noise	Regularization function	MPSNR				
			S [208]	GS [85]	LR [39]	TV [57]	**FC**
Birds1	$[-0.2, 0.2]$	ATV	30.15	30.48	34.85	29.97	**36.53**
		ITV	30.15	30.53	34.26	29.98	**36.53**
		ATV+NN	30.18	30.50	34.76	29.98	**35.28**
	$[-0.25, 0.25]$	ATV	30.02	30.44	32.09	29.98	**36.82**
		ITV	30.02	30.44	33.45	29.99	**36.87**
		ATV+NN	30.06	30.44	33.87	29.99	**35.36**
	$[-0.3, 0.3]$	ATV	31.89	32.36	38.67	31.76	**41.94**
		ITV	31.90	32.37	37.85	31.82	**42.13**
		ATV+NN	31.90	32.37	34.96	31.91	**41.73**
	$[-0.35, 0.35]$	ATV	31.78	32.23	35.84	31.63	**40.70**
		ITV	31.78	32.37	34.88	31.68	**40.54**
		ATV+NN	31.82	32.39	35.69	31.98	**40.69**
	$[-0.4, 0.4]$	ATV	31.43	31.91	34.58	31.19	**39.10**
		ITV	31.43	31.91	34.99	31.27	**39.38**
		ATV+NN	31.37	31.85	34.48	31.36	**38.98**

Table 3.5: MSSIMs of the IR Destriping Results in Case (ii)

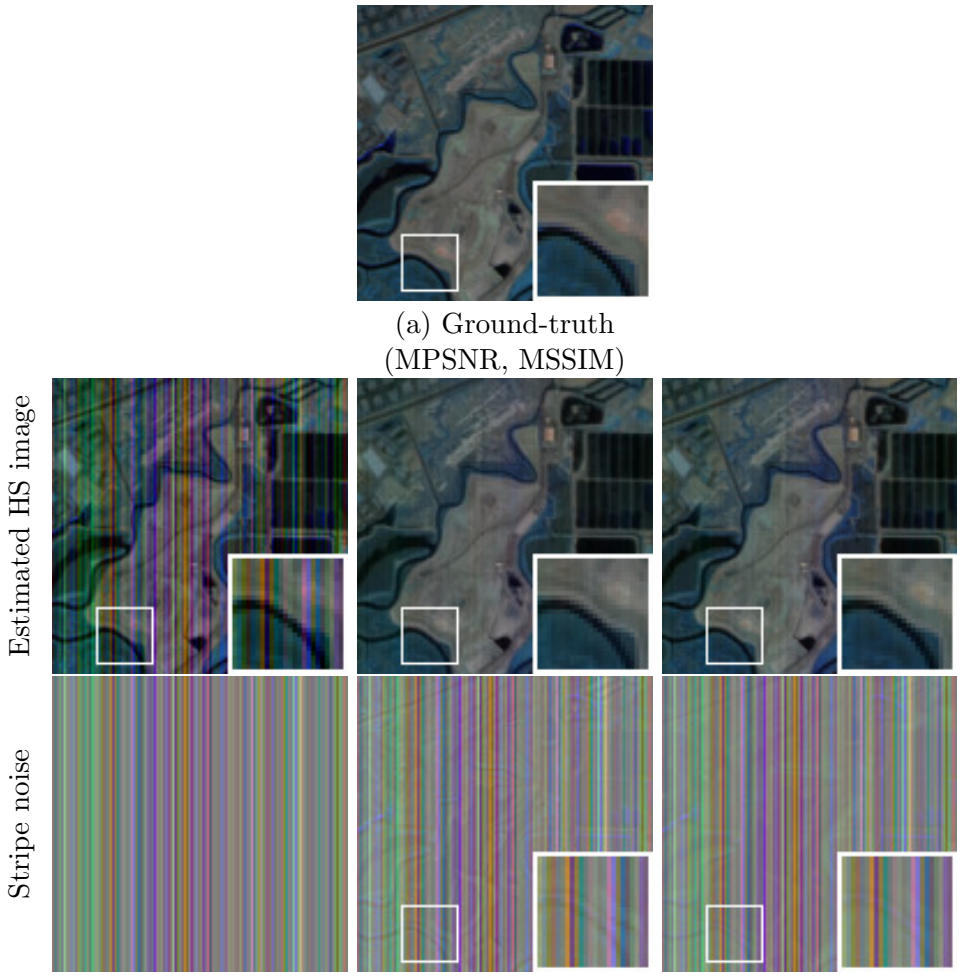
IR video data	Range of stripe noise	Regularization function	MSSIM				
			S [208]	GS [85]	LR [39]	TV [57]	**FC**
Birds1	$[-0.2, 0.2]$	ATV	0.9532	0.9540	0.9955	0.9400	**0.9956**
		ITV	0.9532	0.9524	0.9935	0.9414	**0.9957**
		ATV+NN	0.9531	0.9530	**0.9951**	0.9486	**0.9951**
	$[-0.25, 0.25]$	ATV	0.9526	0.9540	0.9771	0.9399	**0.9956**
		ITV	0.9525	0.9540	0.9936	0.9410	**0.9959**
		ATV+NN	0.9528	0.9538	0.9950	0.9485	**0.9951**
	$[-0.3, 0.3]$	ATV	0.9541	0.9552	0.9945	0.9420	**0.9953**
		ITV	0.9539	0.9551	0.9939	0.9430	**0.9954**
		ATV+NN	0.9539	0.9551	0.9775	0.9504	**0.9953**
	$[-0.35, 0.35]$	ATV	0.9539	0.9548	0.9914	0.9422	**0.9955**
		ITV	0.9539	0.9553	0.9844	0.9424	**0.9958**
		ATV+NN	0.9540	0.9554	0.9909	0.9503	**0.9956**
	$[-0.4, 0.4]$	ATV	0.9537	0.9541	0.9899	0.9378	**0.9954**
		ITV	0.9537	0.9541	0.9840	0.9393	**0.9956**
		ATV+NN	0.9536	0.9540	0.9900	0.9467	**0.9953**

Table 3.6: MPSNRs of the HS Image Destriping Results in Case (iii)

HSI	Range of stripe noise	Regularization function	MPSNR				
			S [208]	GS [85]	LR [39]	TV [57]	**FC**
Salinas	$[-0.2, 0.2]$	HTV	29.16	29.30	31.05	**31.08**	**31.08**
		SSTV	33.55	33.91	**34.60**	34.89	34.49
		ASSTV	28.93	28.98	**29.11**	28.83	28.96
		TNN	24.10	24.35	15.83	**26.52**	**26.52**
		SSTV+TNN	32.38	32.95	34.63	**34.64**	33.84
		l_0-l_1HTV	35.58	35.90	37.10	37.09	**37.17**
	$[-0.25, 0.25]$	HTV	28.74	29.10	30.68	**30.98**	**30.98**
		SSTV	33.01	33.59	34.16	**34.39**	**34.39**
		ASSTV	28.86	28.92	**29.03**	28.79	28.94
		TNN	23.34	23.89	23.67	**26.40**	**26.40**
		SSTV+TNN	31.63	32.49	34.14	**34.39**	33.51
		l_0-l_1HTV	35.12	35.28	36.71	36.91	**36.99**
	$[-0.3, 0.3]$	HTV	28.35	28.97	30.36	**30.94**	**30.94**
		SSTV	32.63	33.39	33.73	**36.14**	34.34
		ASSTV	28.80	28.89	**28.98**	28.77	28.92
		TNN	22.43	23.78	23.15	26.30	**28.30**
		SSTV+TNN	31.14	32.24	33.79	**34.29**	33.41
		l_0-l_1HTV	34.80	34.85	36.35	36.84	**36.92**
	$[-0.35, 0.35]$	HTV	28.03	28.86	30.17	**30.90**	**30.90**
		SSTV	32.30	33.17	33.22	**34.20**	**34.20**
		ASSTV	28.74	28.85	**28.91**	28.75	**28.91**
		TNN	21.50	23.69	22.69	**26.10**	**26.10**
		SSTV+TNN	30.84	32.10	33.58	**34.22**	33.37
		l_0-l_1HTV	34.49	34.50	36.00	36.72	**36.80**
	$[-0.4, 0.4]$	HTV	27.69	28.73	30.12	**30.88**	**30.88**
		SSTV	32.17	33.17	32.99	**34.60**	34.31
		ASSTV	28.70	28.84	28.86	28.73	**28.90**
		TNN	21.55	23.47	15.26	**25.54**	**25.54**
		SSTV+TNN	30.58	31.99	33.25	**34.22**	33.38
		l_0-l_1HTV	34.39	34.31	35.58	36.86	**36.94**
Moffett Field	$[-0.2, 0.2]$	HTV	27.44	28.04	29.01	29.10	**29.15**
		SSTV	33.51	33.90	**34.35**	33.78	**34.35**
		ASSTV	27.08	28.00	28.02	28.00	**28.18**
		TNN	23.91	25.50	21.41	**31.29**	**31.29**
		SSTV+TNN	32.40	33.19	**34.78**	33.85	33.92
		l_0-l_1HTV	33.98	34.60	35.90	35.76	**35.92**
	$[-0.25, 0.25]$	HTV	27.22	27.69	28.82	**29.19**	29.14
		SSTV	33.16	33.68	34.05	34.37	34.37
		ASSTV	27.05	27.96	27.98	27.97	28.17
		TNN	23.85	25.42	28.70	**31.23**	**31.23**
		SSTV+TNN	32.03	32.94	**34.61**	33.79	33.90
		l_0-l_1HTV	33.74	34.20	35.66	35.74	**35.91**
	$[-0.3, 0.3]$	HTV	26.96	27.38	28.56	29.05	**29.12**
		SSTV	32.71	33.36	33.55	33.52	**34.27**
		ASSTV	25.98	26.15	26.04	**27.94**	26.14
		TNN	23.77	25.36	28.18	**30.97**	**30.97**
		SSTV+TNN	31.59	32.57	**34.23**	33.65	33.79
		l_0-l_1HTV	33.39	33.74	35.35	35.63	**35.80**
	$[-0.35, 0.35]$	HTV	26.73	27.11	28.34	**29.15**	29.11
		SSTV	32.60	33.39	33.30	**34.32**	34.31
		ASSTV	27.01	27.92	27.93	27.94	**28.16**
		TNN	23.75	25.39	27.97	**31.38**	**31.38**
		SSTV+TNN	31.34	32.43	**34.18**	33.67	33.79
		l_0-l_1HTV	33.30	33.62	35.14	35.70	**35.87**
	$[-0.4, 0.4]$	HTV	26.40	26.79	27.94	28.93	**29.05**
		SSTV	32.31	33.14	32.85	33.43	**34.19**
		ASSTV	26.98	27.90	27.89	27.93	**28.16**
		TNN	23.72	25.26	20.13	**31.41**	**31.41**
		SSTV+TNN	30.93	32.09	**33.65**	33.52	33.63
		l_0-l_1HTV	33.01	33.19	34.64	35.53	**35.71**

Table 3.7: MSSIMs of the HS Image Destriping Results in Case (iii)

HSI	Range of stripe noise	Regularization function	MSSIM				
			S [208]	GS [85]	LR [39]	TV [57]	**FC**
Salinas	$[-0.2, 0.2]$	HTV	0.8371	0.8371	**0.8717**	0.8698	0.8698
		SSTV	0.8643	0.8770	0.8910	0.8772	**0.8913**
		ASSTV	0.6669	0.6268	0.6473	0.6613	**0.6648**
		TNN	0.5456	0.4218	0.4802	**0.8662**	**0.8662**
		SSTV+TNN	0.8893	0.9153	**0.9382**	0.9209	0.9097
		l_0-l_1HTV	0.9384	0.9401	**0.9523**	0.9480	0.9483
	$[-0.25, 0.25]$	HTV	0.8317	0.8354	**0.8702**	0.8696	0.8696
		SSTV	0.8575	0.8757	0.8882	**0.8917**	**0.8917**
		ASSTV	0.6643	0.6260	0.6471	0.6600	**0.6638**
		TNN	0.4505	0.7146	0.8511	**0.8549**	**0.8549**
		SSTV+TNN	0.8738	0.9103	**0.9370**	0.9194	0.9072
		l_0-l_1HTV	0.9360	0.9357	**0.9518**	0.9474	0.9476
	$[-0.3, 0.3]$	HTV	0.8247	0.8338	0.8688	**0.8695**	**0.8695**
		SSTV	0.8495	0.8727	0.8817	**0.9090**	0.8910
		ASSTV	0.6624	0.6256	0.6468	0.6591	**0.6632**
		TNN	0.3678	0.7004	**0.8421**	0.8311	0.8311
		SSTV+TNN	0.8611	0.9067	**0.9362**	0.9185	0.9058
		l_0-l_1HTV	0.9337	0.9318	**0.9513**	0.9472	0.9474
	$[-0.35, 0.35]$	HTV	0.8166	0.8323	0.8680	**0.8693**	**0.8693**
		SSTV	0.8392	0.8668	0.8649	**0.8881**	**0.888**
		ASSTV	0.6608	0.6251	0.6463	0.6586	**0.6628**
		TNN	0.3066	0.6758	**0.8276**	0.7774	0.7774
		SSTV+TNN	0.8527	0.9047	**0.9349**	0.9179	0.9052
		l_0-l_1HTV	0.9306	0.9281	**0.9501**	0.9468	0.9471
	$[-0.4, 0.4]$	HTV	0.8098	0.8309	0.8599	**0.8688**	**0.8688**
		SSTV	0.8370	0.8671	0.8686	0.8730	**0.8898**
		ASSTV	0.6598	0.6251	0.6460	0.6581	**0.6623**
		TNN	0.6607	0.6198	0.3181	**0.6978**	**0.6978**
		SSTV+TNN	0.8462	0.9029	**0.9347**	0.9173	0.9046
		l_0-l_1HTV	0.9304	0.9266	**0.9500**	0.9471	0.9474
Moffett Field	$[-0.2, 0.2]$	HTV	0.6387	0.6467	0.7043	0.7153	**0.7275**
		SSTV	0.8421	0.8466	**0.8574**	0.8317	0.8548
		ASSTV	0.6118	0.6327	**0.6353**	0.6234	0.6290
		TNN	0.5512	0.6240	0.4698	**0.8581**	**0.8581**
		SSTV+TNN	0.8413	0.8406	**0.8933**	0.8783	0.8698
		l_0-l_1HTV	0.8778	0.8812	**0.9060**	0.9008	0.9022
	$[-0.25, 0.25]$	HTV	0.6269	0.6294	0.6977	**0.7291**	0.7278
		SSTV	0.8339	0.8436	0.8519	**0.8565**	0.8564
		ASSTV	0.6112	0.6320	**0.6339**	0.6222	0.6284
		TNN	0.5477	0.6203	0.7184	**0.8506**	**0.8506**
		SSTV+TNN	0.8312	0.8364	**0.8910**	0.8799	0.8715
		l_0-l_1HTV	0.8738	0.8747	**0.9039**	0.9023	0.9037
	$[-0.3, 0.3]$	HTV	0.6092	0.6125	0.6834	0.7120	**0.7256**
		SSTV	0.8175	0.8317	0.8337	0.8241	**0.8497**
		ASSTV	0.4805	0.4871	0.4838	**0.6195**	0.4870
		TNN	0.5427	0.6138	0.6822	**0.8387**	**0.8387**
		SSTV+TNN	0.8138	0.8235	**0.8820**	0.8754	0.8669
		l_0-l_1HTV	0.8615	0.8600	0.8926	0.8958	**0.8972**
	$[-0.35, 0.35]$	HTV	0.5994	0.6024	0.6790	**0.7279**	0.7271
		SSTV	0.8170	0.8348	0.8309	**0.8548**	0.8542
		ASSTV	0.6087	0.6303	**0.6319**	0.6207	0.6277
		TNN	0.5379	0.6006	0.6643	**0.8403**	**0.8403**
		SSTV+TNN	0.8629	0.8605	0.8932	0.9012	**0.9025**
		l_0-l_1HTV	0.8629	0.8605	0.8932	0.9012	**0.9025**
	$[-0.4, 0.4]$	HTV	0.5807	0.5871	0.6603	0.7071	**0.7210**
		SSTV	0.8069	0.8263	0.8150	0.8222	**0.8472**
		ASSTV	0.6073	0.6280	**0.6283**	0.6184	0.6257
		TNN	0.5359	0.5713	0.3164	**0.8293**	**0.8293**
		SSTV+TNN	0.7939	0.8127	0.8670	**0.8710**	0.8630
		l_0-l_1HTV	0.8504	0.8472	0.8777	0.8926	**0.8939**



(a) Ground-truth (MPSNR, MSSIM)

(b) Observation

(c) S [208] (35.10, 0.8871)

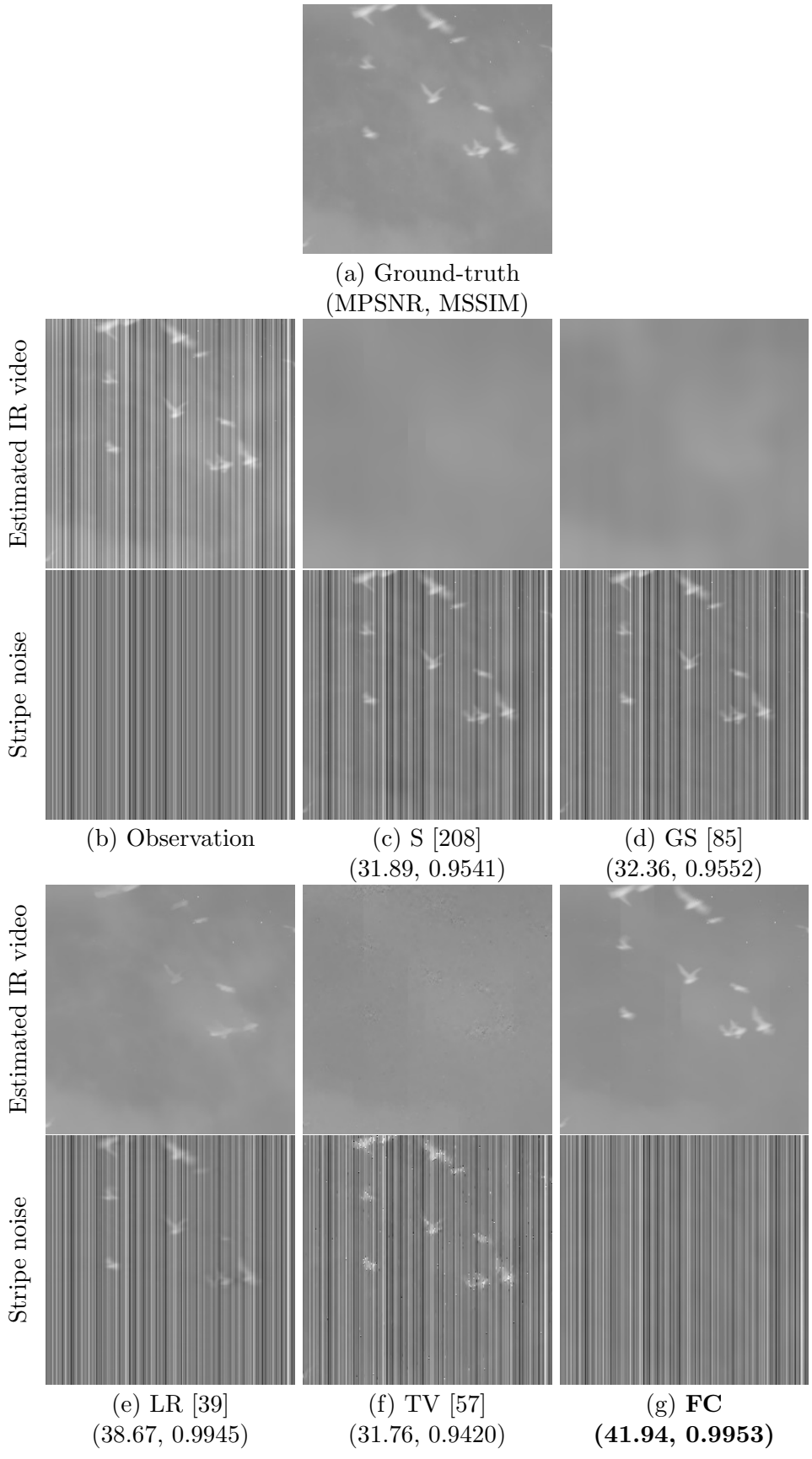
(d) GS [85] (35.79, 0.8919)

(e) LR [39] (36.93, 0.9005)

(f) TV [57] (35.85, 0.8711)

(g) **FC** **(40.84, 0.9548)**

Figure 3.6: *Moffett field* destriping results in Case (i) with SSTV (R: 126, G: 95, B: 74). The MPSNR and MSSIM of our FC are highlighted in bold.



(a) Ground-truth (MPSNR, MSSIM)

(b) Observation

(c) S [208] (31.89, 0.9541)

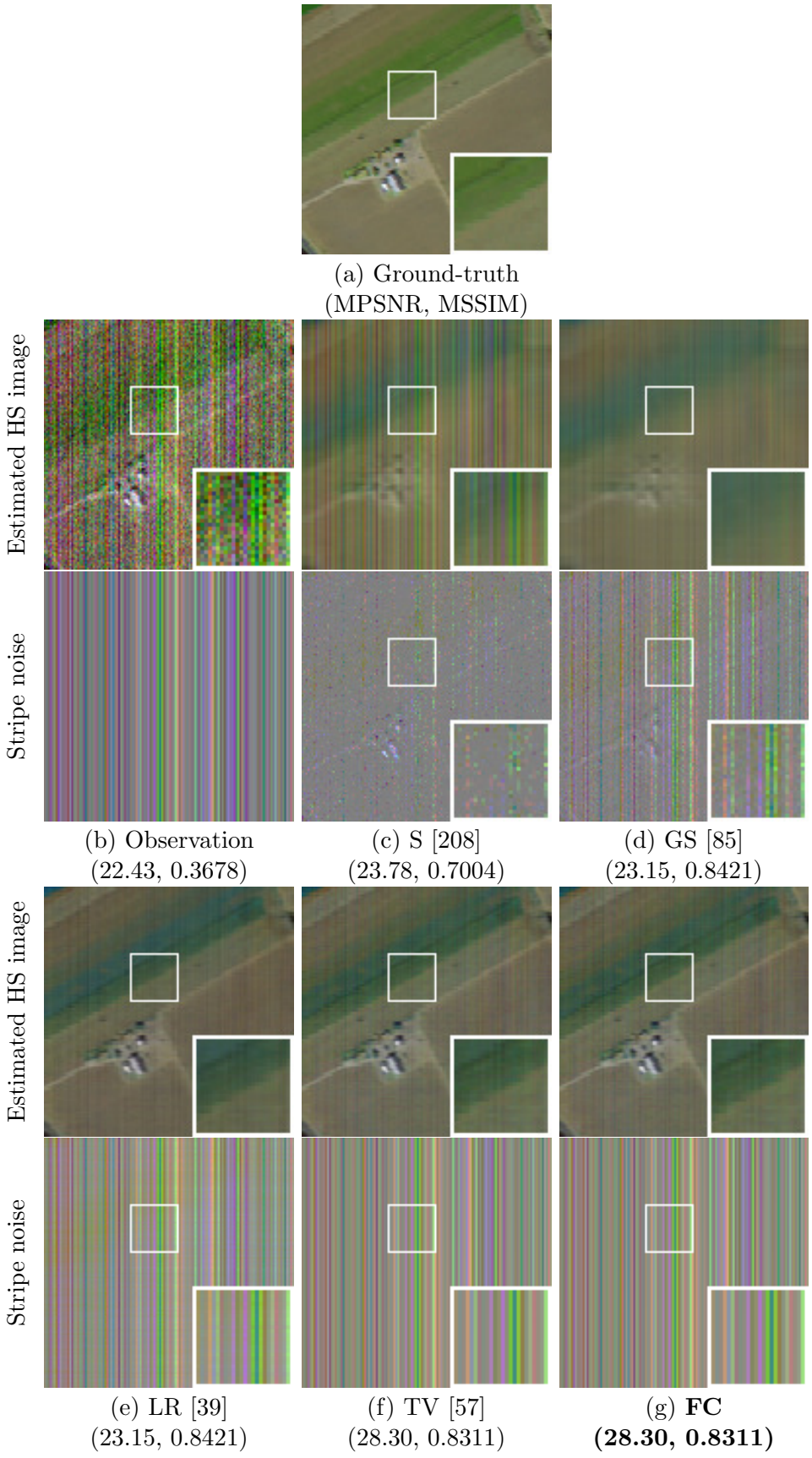
(d) GS [85] (32.36, 0.9552)

(e) LR [39] (38.67, 0.9945)

(f) TV [57] (31.76, 0.9420)

(g) **FC** **(41.94, 0.9953)**

Figure 3.7: *Birds1* destriping results in Case (ii) with ATV. The MPSNR and MSSIM of our FC are highlighted in bold.



(a) Ground-truth
(MPSNR, MSSIM)

(b) Observation
(22.43, 0.3678)

(c) S [208]
(23.78, 0.7004)

(d) GS [85]
(23.15, 0.8421)

(e) LR [39]
(23.15, 0.8421)

(f) TV [57]
(28.30, 0.8311)

(g) **FC**
(28.30, 0.8311)

Figure 3.8: *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.

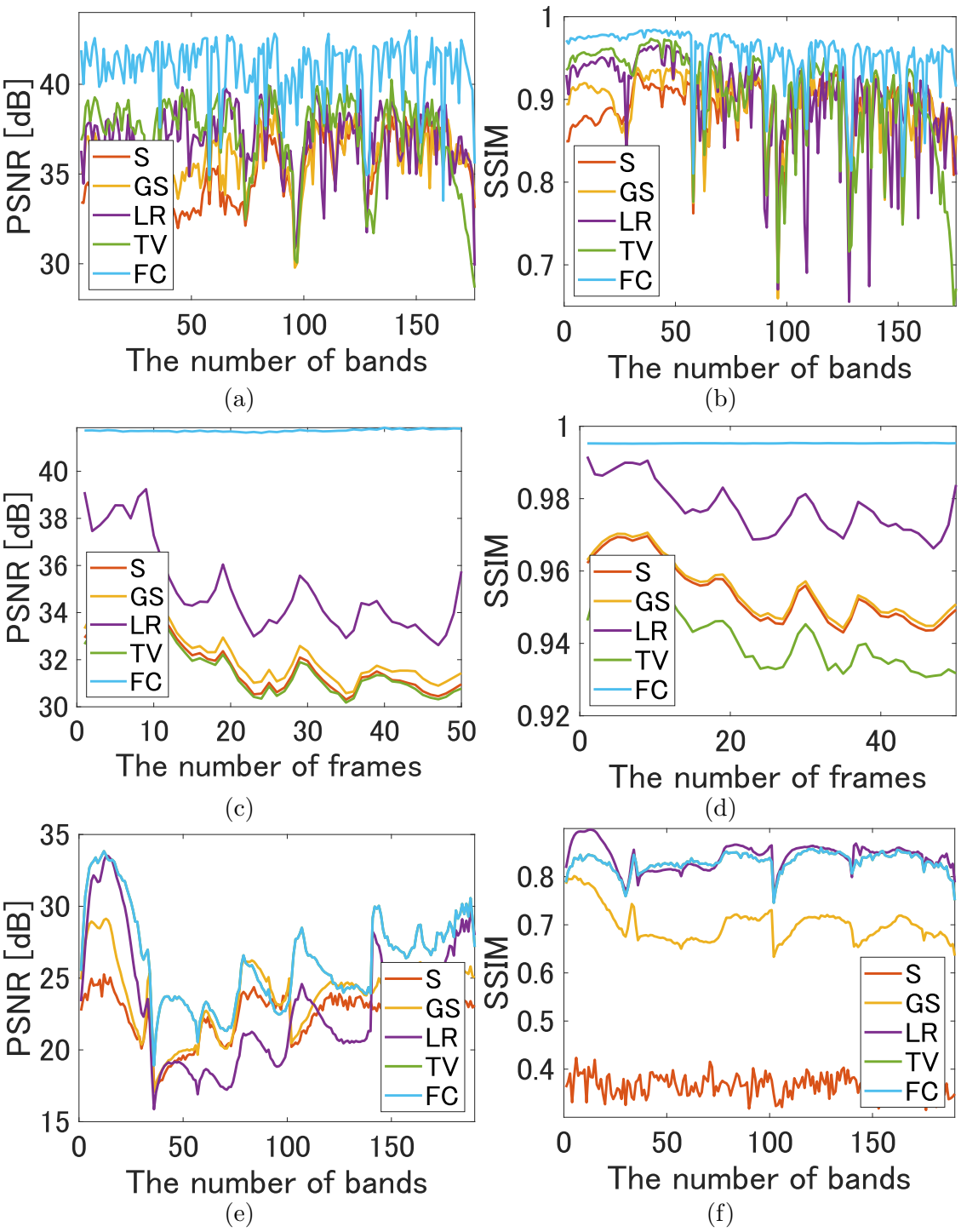


Figure 3.9: Band-wise or frame-wise PSNRs and SSIMs. Note that these are not MPSNRs and MSSIMs. (a) and (b) are the PSNRs and SSIMs of the *Moffett field* destriping results in Case (i) using SSTV. (c) and (d) are the PSNRs and SSIMs of the *Birds1* destriping results in Case (ii) using ATV. (e) and (f) are the PSNRs and SSIMs of the *Salinas* destriping results in Case (iii) using TNN.

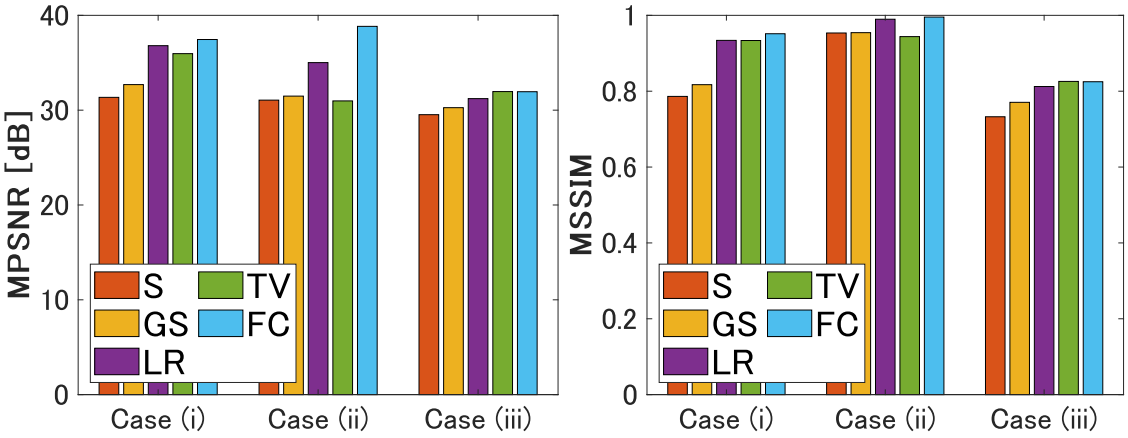


Figure 3.10: The means of MPSNRs and MSSIMs in each noise case.

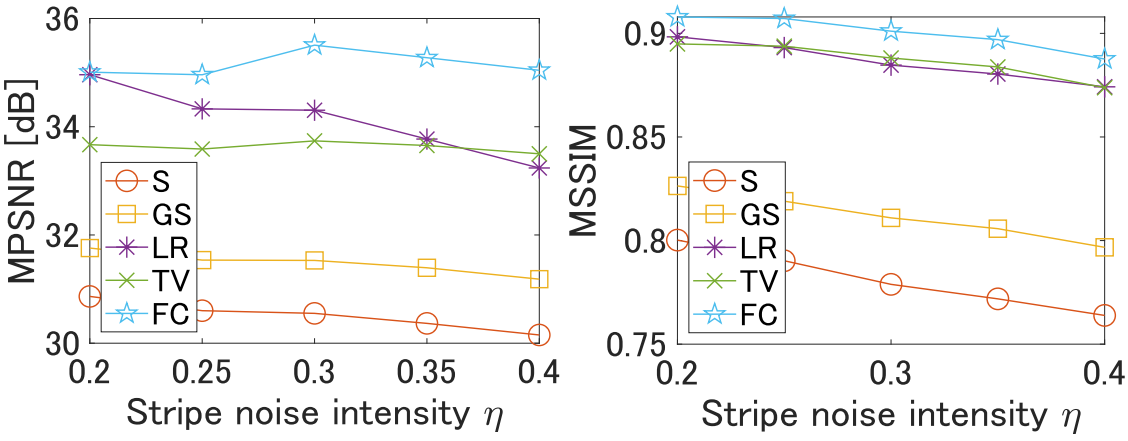


Figure 3.11: The means of MPSNRs and MSSIMs in each stripe noise intensity range $[-\eta, \eta]$.

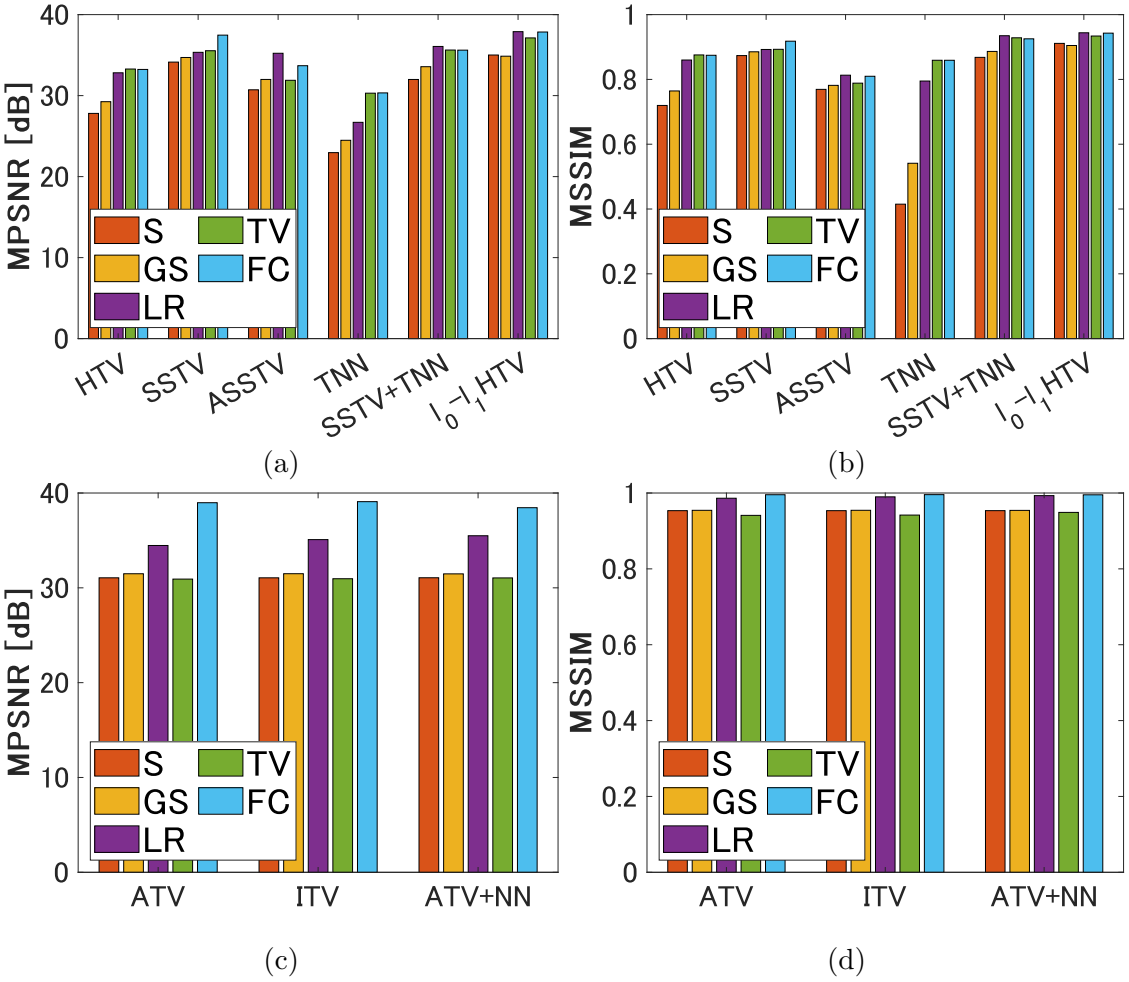
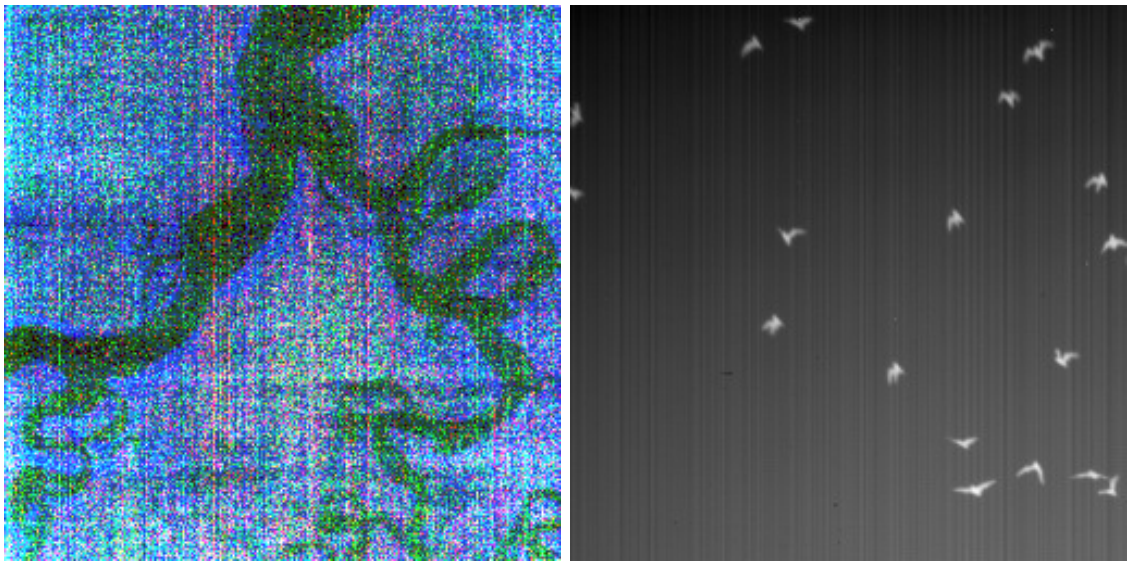


Figure 3.12: The means of MPSNRs and MSSIMs in each image regularization. (a): The means of MPSNRs in the HS image experiments. (b): The means of MSSIMs in the HS image experiments. (c): The means of MPSNRs in the IR video experiments. (d): The means of MSSIMs in the IR video experiments.



(a) (b)

Figure 3.13: HS image and IR video data used for experiments in real noise cases. (a) *Suwannee* (an HS image depicted by a false-color image of R: 357, G: 275, B: 120). (b) *Birds2* (an IR video).

3.4.4 Experiments in Real Noise Cases

In the real noise-case experiments, the parameter λ (Tab. 3.1) 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 ε, 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 $\frac{\|\mathbf{u}^{(t+1)}-\mathbf{u}^{(t)}\|_2}{\|\mathbf{u}^{(t)}\|_2} < 1.0\times10^{-4}$.

We show the *Suwannee* destriping results for a real noise case in Figs. 3.14 and 3.15. The destriping result by S-HTV (Fig. 3.14 (a1)) includes residual stripe noise. The results by S-SSTV (Fig. 3.14 (a2)), GS-SSTV (Fig. 3.14 (b2)), S-ASSTV (Fig. 3.14 (a3)), GS-ASSTV (Fig. 3.15 (b3)), S-TNN (Fig. 3.15 (a4)), GS-TNN (Fig. 3.15 (b4)), and S-l_0-l_1HTV (Fig. 3.15 (a6)) have brighter areas than the original image (Fig. 3.13 (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 (Fig. 3.14 (c3)) recovered the narrow river that lies along with the vertical direction in the magnified areas. On the other hand, LR-SSTV (Fig. 3.14 (c2)) and LR-TNN (Fig. 3.15 (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 (Fig. 3.14 (d2)), TV-ASSTV (Fig. 3.14 (d3)), TV-SSTV+TNN (Fig. 3.15 (d5)), and TV-l_0-l_1HTV (Fig. 3.15 (d6)), the land shape was also partially removed as stripe noise. For example, TV-ASSTV (Fig. 3.14 (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-l_1HTV, 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, FC-l_0-l_1HTV, its strong ability of stripe noise characterization allows us to achieve desirable destriping. However, our results do not satisfy the flatness constraint and slightly include land shapes in the stripe noise components only for FC-ASSTV and FC-SSTV+TNN (Figs. 3.14 (e3) and 3.15 (e5)). This indicates that FC-

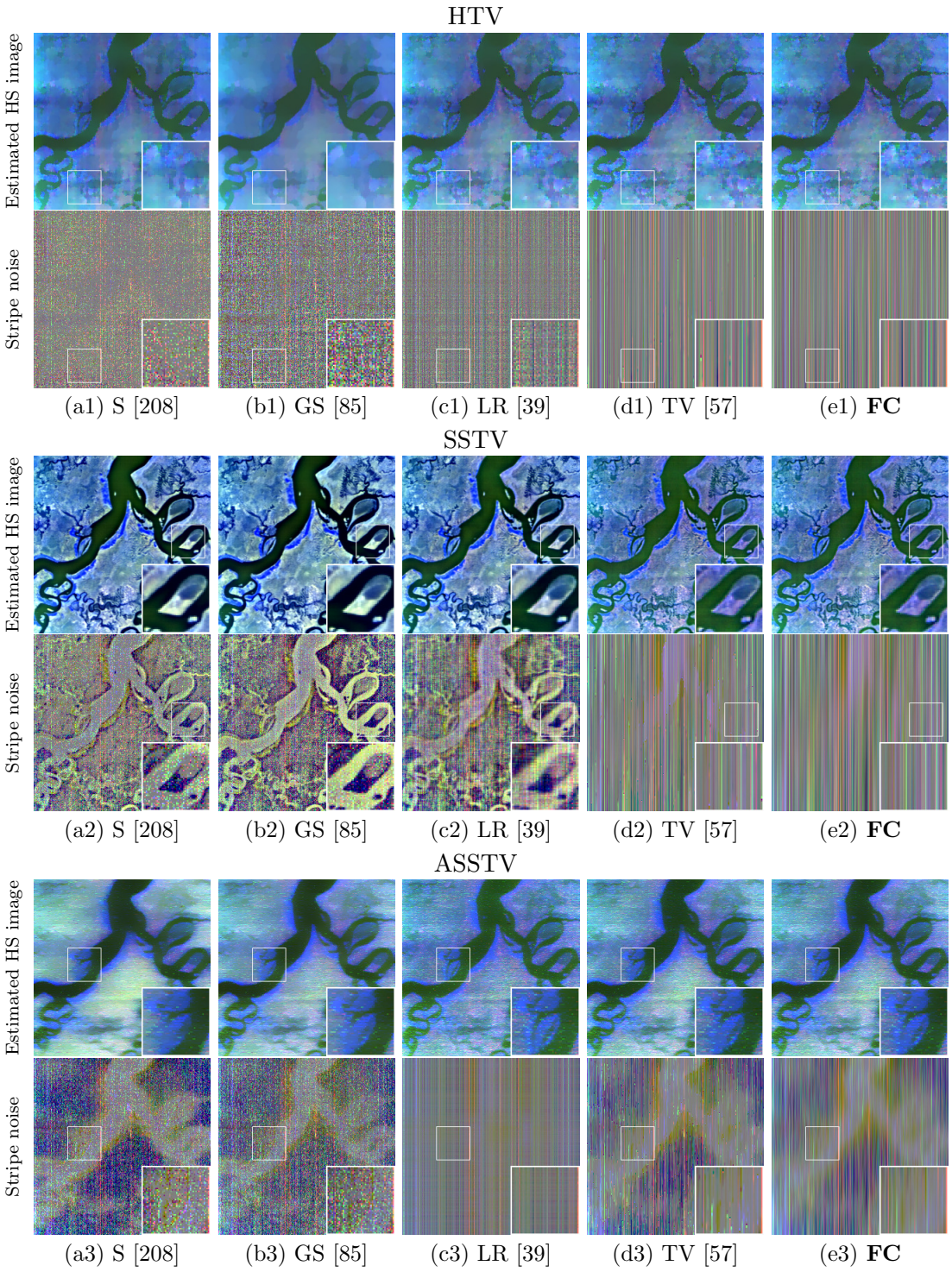


Figure 3.14: HSI destriping results in real noise cases (R: 357, G: 275, B: 120). The top rows and bottom rows are the estimated HS images and the estimated stripe noise, respectively.

ASSTV and FC-SSTV+TNN need more iterations to preclude the land shapes from their stripe noise components.

Figure 3.16 shows the destriping results of the IR video *Bats2*. S and TV removed bats (moving objects) as stripe noise. This is because the stripe noise components (Figs. 3.16 (a1), 3.16 (a2), 3.16 (a3), 3.16 (d1), 3.16 (d2), and 3.16 (d3)) have sparse or vertical smoothness properties. GS and LR performed better than S and TV, but some of the

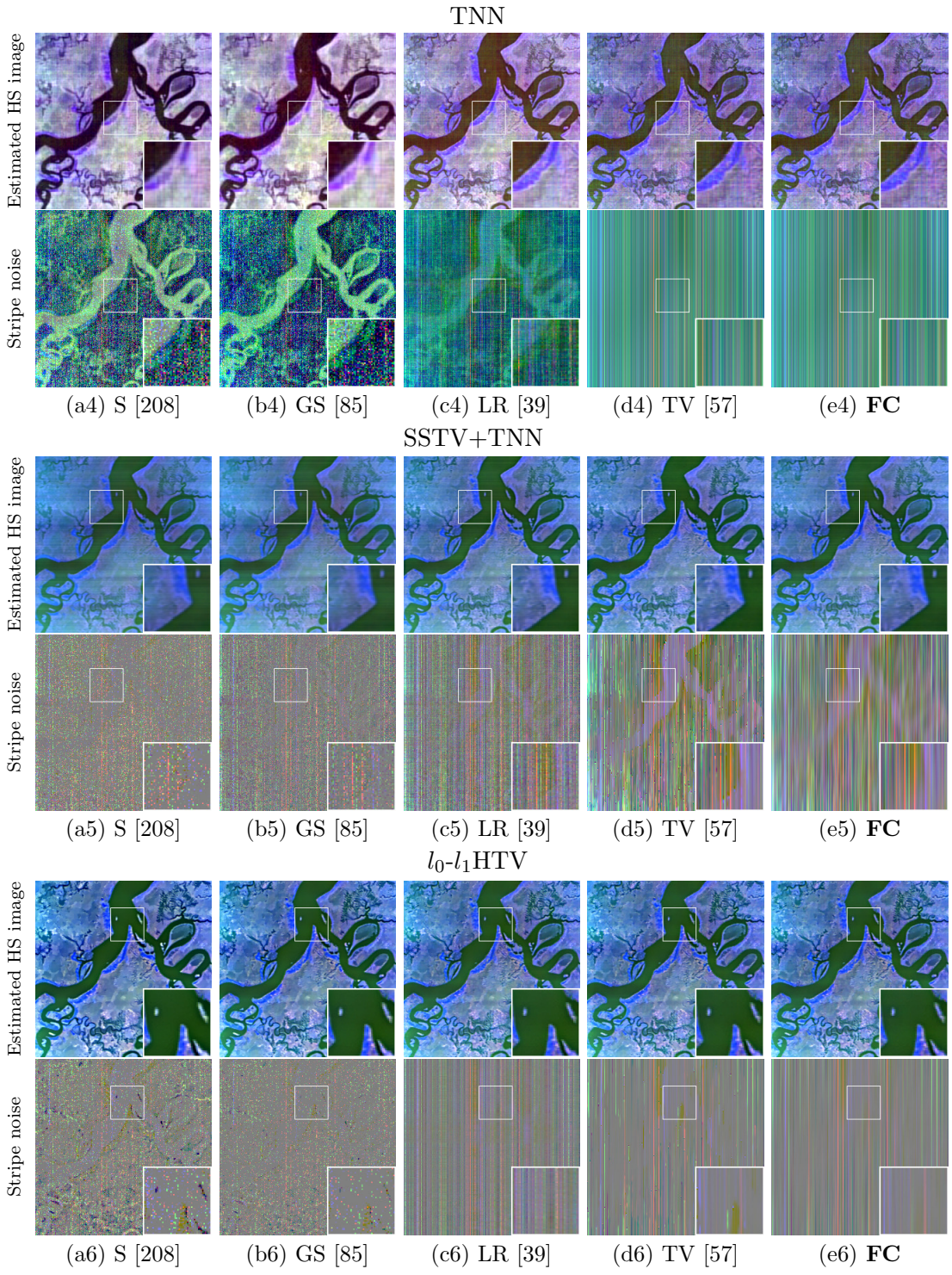


Figure 3.15: HSI destriping results in real noise cases (R: 357, G: 275, B: 120) using TNN, SSTV+TNN, and l_0-l_1HTV. The top rows and bottom rows are the estimated HS images and the estimated stripe noise, respectively.

bats were regarded as stripe noise components (see Figs. 3.16 (b1), 3.16 (b2), 3.16 (b3), 3.16 (c1), 3.16 (c2), and 3.16 (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 the bats (see Figs. 3.16 (e1), 3.16 (e2), and 3.16 (e3)).

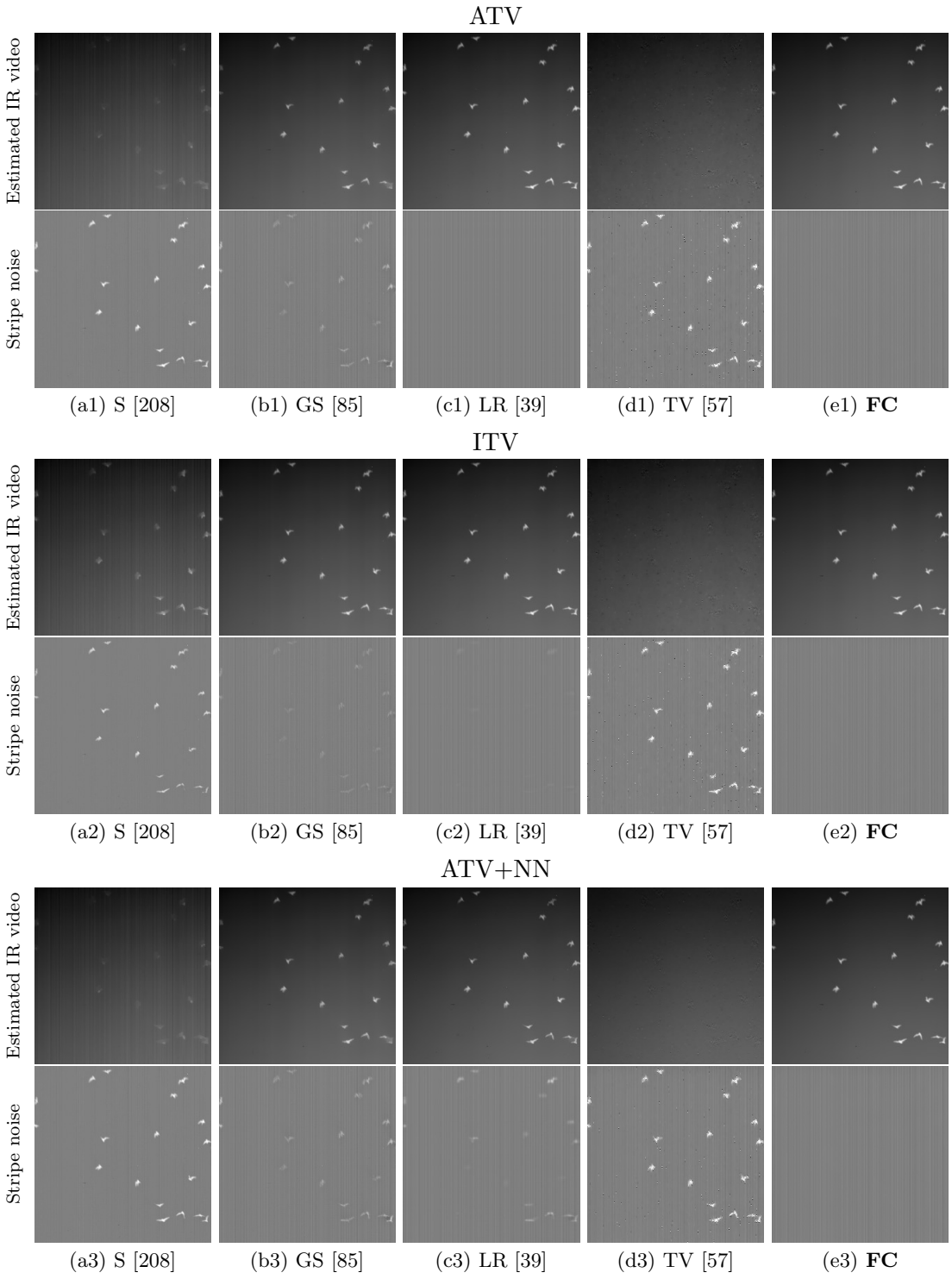
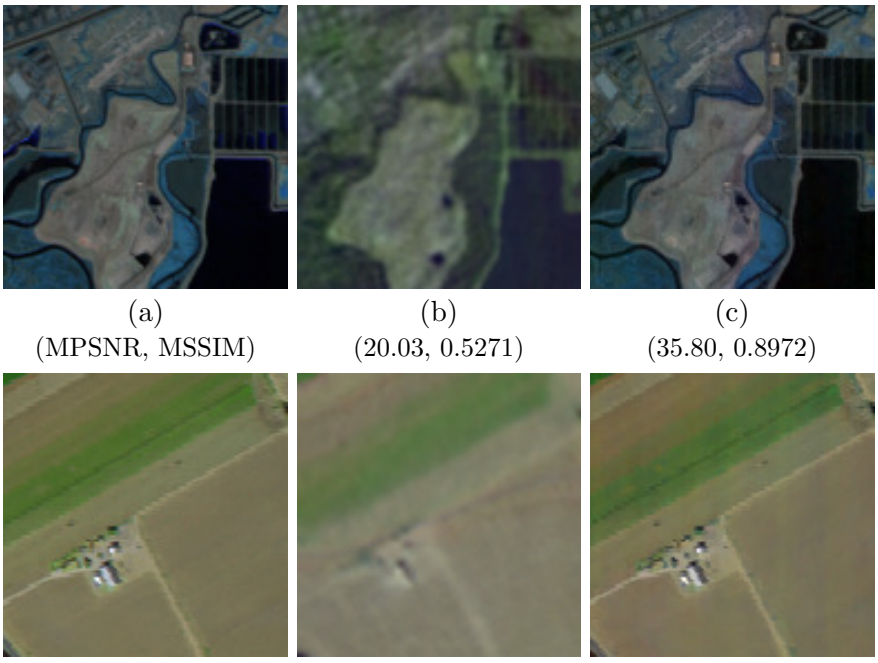


Figure 3.16: IR video destriping results in real noise cases.

3.4.5 Comparison With A Deep Learning-Based Method

We compare our framework with a deep learning-based method [166] [1], 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. Figure 3.17 shows the destriping results, which validate the effectiveness of our framework compared to a deep learning-based method. The method in [166] did not

[1]The code is available at `https://github.com/acecreamu/deep-hs-prior`.



(a) (MPSNR, MSSIM) (b) (20.03, 0.5271) (c) (35.80, 0.8972)

(d) (MPSNR, MSSIM) (e) (23.71, 0.8059) (f) (36.92, 0.9474)

Figure 3.17: Comparison with a deep learning-based method [166]. (a) and (d) are ground-truth images of the *Moffett Field* and *Salinas*, respectively. (b) and (e) are denoising results of [166]. (c) and (f) are denoising results of our framework (FC-l_0-l_1HTV).

recover edges and objects (Figs. 3.17 (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 [92, 115].

3.4.6 Discussion

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

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

3.5 Concluding Remarks

In this paper, we have proposed a general destriping framework for remote sensing images. Specifically, we formulated the destriping as a convex optimization problem equipped with the flatness constraint. Thanks 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

stepsize adjustment. Through destriping experiments using HS image 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 an extension to consider the various degradations such as the spectral variability and the effectiveness demonstration in remote sensing image applications such as classification, unmixing, compressed sensing reconstruction, and target recognition.

Chapter 4

Problem Structure-Based Stepsize Design via Variable-Wise Diagonal Preconditioning for Primal-Dual Splitting

4.1 Introduction

Many signal estimation and processing problems, such as denoising, interpolation, decomposition, and reconstruction, have been resolved by casting them as convex optimization problems [48, 146] of the form in (2.28). Here, we write down it again:

$$\min_{\substack{\mathbf{x}_1,\ldots,\mathbf{x}_N,\\ \mathbf{y}_1,\ldots,\mathbf{y}_M}} \sum_{i=1}^{N} f_i(\mathbf{x}_i) + \sum_{j=1}^{M} g_j\left(\mathbf{y}_j\right) \text{ s.t. } \begin{cases} \mathbf{y}_1 = \sum_{i=1}^{N} \mathfrak{L}_{1,i}(\mathbf{x}_i), \\ \vdots, \\ \mathbf{y}_M = \sum_{i=1}^{N} \mathfrak{L}_{M,i}(\mathbf{x}_i), \end{cases} \tag{4.1}$$

where $f_i : \mathbb{R}^{n_i} \to (-\infty, +\infty]$ and $g_j : \mathbb{R}^{m_j} \to (-\infty, +\infty]$ are proximable proper lower-semicontinuous convex functions, and $\mathfrak{L}_{j,i} : \mathbb{R}^{n_i} \to \mathbb{R}^{m_j}$ is a linear operator ($\forall i = 1, \ldots, N$ and $\forall j = 1, \ldots, M$). The variables $\mathbf{x}_1, \ldots, \mathbf{x}_N$ represent estimated signals or components, and $\mathbf{y}_1, \ldots, \mathbf{y}_M$ are auxiliary variables for splitting.

As a method for solving Prob. (4.1), a primal-dual splitting method (PDS) [32] has attracted attention [35, 52, 70, 73, 97, 101, 127, 137, 140] due to its simple implementation without operator inversions.[1] To improve the convergence speed of PDS, a preconditioned PDS (P-PDS) has been studied [118, 148, 189, 202]. P-PDS is a generalization of the standard PDS, where the scalar-valued stepsizes of PDS are replaced by (positive definite) matrix-valued preconditioners. The theoretical convergence of P-PDS is established in a primal-dual space equipped with a skewed metric, which is determined by the linear operators involved in the optimization problem and the preconditioners used (see [51, 141, 148] for details). Preconditioning can be viewed as the selection of an appropriate metric for optimization algorithms and is a crucial long-standing issue not only in P-PDS but also in various proximal algorithms [14, 50].

The appropriate preconditioners that accelerate the convergence of P-PDS vary greatly depending on the structure of the target optimization problem (see Section 3.4 for detailed

[1] This algorithm has been generalized by Condat [51] and Vu [182], where smooth convex functions are optimized by using their Lipschitzian gradients.

examples). To automatically determine such preconditioners, the authors in [148] have proposed a diagonal-preconditioner design method. The elements of the diagonal preconditioners consist of the row/column absolute sum of the elements of the explicit matrices representing the linear operators $\mathfrak{L}_{j,i}$ in (4.1), and thus the resulting diagonal elements of the preconditioners can be different for each element in one variable.

Although this design method determines reasonable diagonal preconditioners, there exist two limitations that are considerable in real-world applications. First, the method is difficult to apply in the case where (some of) the linear operators $\mathfrak{L}_{j,i}$ in Prob. (4.1) are not implemented as explicit matrices because it requires access to the entire elements of the matrices to construct the preconditioners. We often encounter such situations, especially in imaging applications, where the linear operators are implemented not as explicit matrices but as procedures that compute forward and adjoint operations in an efficient manner, e.g., difference operators [22, 31] and frame transforms [24, 98, 145]. Second, some proximable functions f_i and g_j are not completely separable for each element of the input variables $\mathbf{x}_i$ and $\mathbf{y}_j$, e.g., mixed norms and the indicator functions of norm balls [45]. For such functions, the element-wise preconditioning might make the functions non-proximable.

To address the above issues, we propose an Operator-norm-based design method of Variable-wise Diagonal Preconditioning (OVDP). Specifically, we introduce a new general form of P-PDS preconditioners, and then propose specific preconditioners based on this general form. We also prove that the sequence generated by P-PDS with OVDP converges to an optimal solution of Prob. (4.1).

Our method has two features preferred in many real-world applications. First, our preconditioners can be computed from (upper bounds of) the operator norms of the linear operators $\mathfrak{L}_{j,i}$, meaning that our method does not need their explicit matrix representations. This is because (upper bounds of) the operator norms are often known or can be estimated without matrix implementation for typical linear operators used in signal processing applications, including the ones mentioned above. Second, the elements of the diagonal preconditioners obtained by our method take the same value for all the elements of each variable, i.e., variable-wise preconditioning. This maintains the proximablity of the functions f_i and g_j in Prob. (4.1).

Comprehensive experiments are conducted by applying our method to three signal estimation problems: mixed noise removal of hyperspectral images, hyperspectral unmixing, and graph signal recovery. By discussing the convergence in these three optimization problems, which have very different structures, we demonstrate the effectiveness and usefulness of our method.

This chapter is organized as follows. Section 4.2 gives reviews of existing preconditioner design methods. In Section 4.3, we present OVDP and prove the convergence theorem of P-PDS with OVDP. Their applications to mixed noise removal hyperspectral images, hyperspectral unmixing, and graph signal recovery are given in Section 4.4. Finally, we conclude the chapter in Section 4.5.

4.2 Existing Preconditioner Design Methods

4.2.1 Scalar Preconditioning (SP)

The standard PDS [32] can be recovered by setting the preconditioners to be scalar matrices, i.e.,

$$\mathbf{\Gamma}_1 = \gamma_1 \mathbf{I}, \mathbf{\Gamma}_2 = \gamma_2 \mathbf{I}. \tag{4.2}$$

The parameters γ_1 and γ_2 are positive scalars that satisfy (2.32), that is,

$$\gamma_1 \gamma_2 \left\| \mathfrak{L} \right\|_{\mathrm{op}}^2 < 1. \tag{4.3}$$

In practice, the parameter γ_2 is often set as

$$\gamma_2 = \frac{1}{\mu_{SP}^2 \gamma_1}, \tag{4.4}$$

where μ_{SP} is an upper bound of $\|\mathfrak{L}\|_{\mathrm{op}}$. Since $\|\mathfrak{L}\|_{\mathrm{op}} < \mu_{SP}$, the parameters γ_1 and γ_2 in (4.4) satisfy the inequality in (4.3). We note that the parameter γ_1 needs to be manually adjusted for accelerating the convergence of P-PDS.

4.2.2 Row/Column Absolute Sum-Based Element-Wise Preconditioning (ASP)

Let $\mathbf{L}_{j,i}$ be the representation matrix of $\mathfrak{L}_{j,i}$. The authors of [148] present a design method for constructing the preconditioners $\mathbf{\Gamma}_1 = \mathrm{diag}(\mathbf{\Gamma}_{1,1}, \ldots, \mathbf{\Gamma}_{1,N})$ and $\mathbf{\Gamma}_2 = \mathrm{diag}(\mathbf{\Gamma}_{2,1}, \ldots, \mathbf{\Gamma}_{2,M})$ as follows:

$$\begin{aligned}
\mathbf{\Gamma}_{1,i} =& \mathrm{diag}\left(\frac{1}{\sigma_{i,1}}, \ldots, \frac{1}{\sigma_{i,n_i}}\right), \ (\forall i = 1, \ldots, N), \\
\mathbf{\Gamma}_{2,j} =& \mathrm{diag}\left(\frac{1}{\tau_{j,1}}, \ldots, \frac{1}{\tau_{j,m_j}}\right), \ (\forall j = 1, \ldots, M),
\end{aligned} \tag{4.5}$$

where

$$\begin{aligned}
\sigma_{i,l} =& \sum_{j=1}^{M} \sum_{k=1}^{m_j} |[\mathbf{L}_{j,i}]_{k,l}|, \ (\forall l = 1, \ldots, n_i), \\
\tau_{j,l} =& \sum_{i=1}^{N} \sum_{k=1}^{n_i} |[\mathbf{L}_{j,i}]_{l,k}|, \ (\forall l = 1, \ldots, m_j).
\end{aligned} \tag{4.6}$$

Each $\mathbf{\Gamma}_{1,i}$ (or $\mathbf{\Gamma}_{2,j}$) is a diagonal matrix consisting of the row/column absolute sums of the elements of $\mathbf{L}_{j,i}$ (see [148, Lemma 2]). This means that the diagonal elements of one $\mathbf{\Gamma}_{1,i}$ (and $\mathbf{\Gamma}_{2,j}$) may take different values, i.e., the diagonal elements of the preconditioners will be different for each element for one variable in (4.1).

4.2.3 Positive-Definite Preconditioning (PDP)

The authors in [118] proposed to determine the preconditioners as

$$\mathbf{\Gamma}_1 = \tau \mathbf{I}, \mathbf{\Gamma}_2 = \frac{1}{\tau}(\mathbf{L}\mathbf{L}^\top + \theta \mathbf{I})^{-1}, \tag{4.7}$$

where $\mathbf{L}$ is the representation matrix of $\mathfrak{L}$ and $\tau > 0$ is a parameter. Since the preconditioners in (4.7) are not block-diagonal matrices in general, P-PDS with them results in the procedures given in (2.30).

If the number of dual variables is two ($M = 2$), the preconditioners are set as

$$\mathbf{\Gamma}_1 = \frac{\tau}{2}\mathbf{I}, \ \mathbf{\Gamma}_2 = \begin{bmatrix} \mathbf{\Gamma}_{2,1} & \mathbf{O} \\ \mathbf{O} & \mathbf{\Gamma}_{2,2} \end{bmatrix}, \tag{4.8}$$

where

$$\mathbf{\Gamma}_{2,j} = \frac{1}{\tau}\left(\sum_{i=1}^{N} \mathbf{L}_{j,i}\mathbf{L}_{j,i}^\top + \theta \mathbf{I}\right)^{-1}, \ (\forall j = 1, 2). \tag{4.9}$$

Since $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ in (4.8) are block-diagonal matrices, P-PDS with them can solve the Prob. (4.1) by the procedures given in (2.31).

We note that the parameters τ and θ affect the convergence speed of P-PDS. Therefore, the parameters τ and θ need to be manually adjusted.

4.3 Proposed Operator Norm-Based Variable-Wise Diagonal Preconditioning (OVDP)

This section is devoted to the establishment of a novel diagonal preconditioning method, OVDP, for P-PDS. First, we introduce a general form of our preconditioners as follows: for all $i = 1, \ldots, N$ and $j = 1, \ldots, M$

$$\begin{aligned} \boldsymbol{\Gamma}_{1,i} &= \Gamma_{1,i}\mathbf{I} = \frac{1}{\sum_{j=1}^{M} \mu_{j,i}^{2-\beta}}\mathbf{I}, \\ \boldsymbol{\Gamma}_{2,j} &= \Gamma_{2,j}\mathbf{I} = \frac{1}{\sum_{i=1}^{N} \mu_{j,i}^{\beta}}\mathbf{I}, \ (\beta \in [0,2]) \end{aligned} \tag{4.10}$$

where each $\mu_{j,i}$ is an upper bound of the operator norm of each $\mathfrak{L}_{j,i}$, i.e.,

$$\mu_{j,i} \in [\|\mathfrak{L}_{j,i}\|_{\mathrm{op}}, \infty). \tag{4.11}$$

By changing the choice of β, OVDP gives three design ways.

- If we choose $\beta = 0$, the preconditioners by OVDP (OVDP1) become

$$\boldsymbol{\Gamma}_{1,i} = \frac{1}{\sum_{j=1}^{M} \mu_{j,i}^{2}}\mathbf{I}, \ \boldsymbol{\Gamma}_{2,j} = \frac{1}{N}\mathbf{I}. \tag{4.12}$$

- If we choose $\beta = 1$, the preconditioners by OVDP (OVDP2) become

$$\boldsymbol{\Gamma}_{1,i} = \frac{1}{\sum_{j=1}^{M} \mu_{j,i}}\mathbf{I}, \ \boldsymbol{\Gamma}_{2,j} = \frac{1}{\sum_{i=1}^{N} \mu_{j,i}}\mathbf{I}. \tag{4.13}$$

- If we choose $\beta = 2$, the preconditioners by OVDP (OVDP3) become

$$\boldsymbol{\Gamma}_{1,i} = \frac{1}{M}\mathbf{I}, \ \boldsymbol{\Gamma}_{2,j} = \frac{1}{\sum_{i=1}^{N} \mu_{j,i}^{2}}\mathbf{I}, \tag{4.14}$$

Remark 4.3.1 (Two Features of Our Method).

- Our preconditioners can be calculated by only using (upper bounds of) the operator norms of the linear operators $\mathfrak{L}_{j,i}$. This implies that OVDP does not require direct access to the elements of the explicit matrices representing $\mathfrak{L}_{j,i}$ as long as some $\mu_{i,j}$ are available.
- In addition, the diagonal elements of one $\boldsymbol{\Gamma}_{1,i}$ take the same value ($\boldsymbol{\Gamma}_{2,j}$ as well), i.e., our method is a variable-wise preconditioning method, which maintains the proximability of the functions in Prob. (4.1).

Before showing the convergence theorem of P-PDS with OVDP defined in (4.10), we give the following lemma on matrix decomposition.

Lemma 4.3.1. An arbitrary matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be decomposed into matrices $\mathbf{B}$ and $\mathbf{C}$ (i.e., $\mathbf{A} = \mathbf{BC}$) that satisfy for any $\beta \in [0, 1]$

$$\begin{aligned} \|\mathbf{B}\|_{\mathrm{op}} &= \|\mathbf{A}\|_{\mathrm{op}}^{1-\beta} \ (= \sigma_1(\mathbf{A})^{1-\beta}), \\ \|\mathbf{C}\|_{\mathrm{op}} &= \|\mathbf{A}\|_{\mathrm{op}}^{\beta} \ (= \sigma_1(\mathbf{A})^{\beta}). \end{aligned} \tag{4.15}$$

The proof is in Appendix.

Then, the following theorem guarantees the convergence of P-PDS with OVDP.

Algorithm 2 P-PDS with OVDP for solving (4.1)

Input: $\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_N^{(0)}, \mathbf{y}_1^{(0)}, \ldots, \mathbf{y}_M^{(0)}$
Output: $\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)}, \mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_M^{(t)}$

Initialize $t = 0$;
Set $\mathbf{\Gamma}_{1,1}, \ldots, \mathbf{\Gamma}_{1,N}, \mathbf{\Gamma}_{2,1}, \ldots, \mathbf{\Gamma}_{2,M}$ as in (4.10);
while A stopping criterion is not satisfied **do**
    for $i = 1, \cdots, N$ **do**
        $\mathbf{x}_i' \leftarrow \sum_{j=1}^{M} \mathfrak{L}_{j,i}^*(\mathbf{y}_j^{(t)})$;
        $\mathbf{x}_i^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{1,i}^{-1}, f_i}(\mathbf{x}_i^{(t)} - \mathbf{\Gamma}_{1,i}\mathbf{x}_i')$;
    end for
    for $j = 1, \cdots, M$ **do**
        $\mathbf{y}_j' \leftarrow \sum_{i=1}^{N} \mathfrak{L}_{j,i}(2\mathbf{x}_i^{(t+1)} - \mathbf{x}_i^{(t)})$;
        $\mathbf{y}_j^{(t+1)} \leftarrow \mathrm{prox}_{\mathbf{\Gamma}_{2,j}^{-1}, g_j^*}(\mathbf{y}_j^{(t)} + \mathbf{\Gamma}_{2,j}\mathbf{y}_j')$;
    end for
    $t \leftarrow t + 1$;
end while

Theorem 4.3.2. *If the preconditioners are set as* (4.10)*, then the following inequality holds:*

$$\left\| \mathbf{\Gamma}_2^{\frac{1}{2}} \circ \mathfrak{L} \circ \mathbf{\Gamma}_1^{\frac{1}{2}} \right\|_{\mathrm{op}}^2 \leq 1. \tag{4.16}$$

Proof. Since $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ are positive-definite and diagonal, their powers of one-half are

$$\begin{aligned} \mathbf{\Gamma}_1^{\frac{1}{2}} &= \mathrm{diag}\left(\mathbf{\Gamma}_{1,1}^{\frac{1}{2}}, \ldots, \mathbf{\Gamma}_{1,N}^{\frac{1}{2}} \right), \\ \mathbf{\Gamma}_2^{\frac{1}{2}} &= \mathrm{diag}\left(\mathbf{\Gamma}_{2,1}^{\frac{1}{2}} \ldots, \mathbf{\Gamma}_{2,M}^{\frac{1}{2}} \right). \end{aligned} \tag{4.17}$$

By matrix multiplication and Eq. (4.17), we have

$$\mathbf{\Gamma}_2^{\frac{1}{2}} \circ \mathfrak{L} \circ \mathbf{\Gamma}_1^{\frac{1}{2}} = \left[\mathbf{\Gamma}_{2,j}^{\frac{1}{2}} \circ \mathfrak{L}_{j,i} \circ \mathbf{\Gamma}_{1,i}^{\frac{1}{2}} \right]_{1 \leq i \leq N, 1 \leq j \leq M}. \tag{4.18}$$

For all $\mathbf{x} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_N^\top]^\top \in \mathbb{R}^{\tilde{n}}$, the triangle inequality yields

$$\left\| \mathbf{\Gamma}_2^{\frac{1}{2}} \circ \mathfrak{L} \circ \mathbf{\Gamma}_1^{\frac{1}{2}} \mathbf{x} \right\|_2^2 \leq \sum_{j=1}^{M} \sum_{i=1}^{N} \left\| \mathbf{\Gamma}_{2,j}^{\frac{1}{2}} \circ \mathfrak{L}_{j,i} \circ \mathbf{\Gamma}_{1,i}^{\frac{1}{2}} \mathbf{x}_i \right\|_2^2. \tag{4.19}$$

Since $\mathfrak{L}_{j,i}$ $(i = 1, \ldots, N, j = 1, \ldots, M)$ can be represented by matrices, from Lemma 4.3.1, there exist linear operators $\mathfrak{L}_{j,i}^{\frac{\beta}{2}}$ and $\mathfrak{L}_{j,i}^{1-\frac{\beta}{2}}$ that satisfy for any $\beta \in [0, 2]$,

$$\begin{aligned} \mathfrak{L}_{j,i} &= \mathfrak{L}_{j,i}^{1-\frac{\beta}{2}} \circ \mathfrak{L}_{j,i}^{\frac{\beta}{2}}, \\ \|\mathfrak{L}_{j,i}^{1-\frac{\beta}{2}}\|_{\mathrm{op}} &= \|\mathfrak{L}_{j,i}\|_{\mathrm{op}}^{1-\frac{\beta}{2}}, \\ \|\mathfrak{L}_{j,i}^{\frac{\beta}{2}}\|_{\mathrm{op}} &= \|\mathfrak{L}_{j,i}\|_{\mathrm{op}}^{\frac{\beta}{2}}. \end{aligned} \tag{4.20}$$

Thus, it follows, from Eq. (4.20) and the definition and the submultiplicity of operator norms, that

$$\begin{aligned}\text{Eq. (4.19)} &= \sum_{j=1}^{M}\sum_{i=1}^{N}\left\|\boldsymbol{\Gamma}_{2,j}^{\frac{1}{2}}\circ\mathfrak{L}_{j,i}^{1-\frac{\beta}{2}}\circ\mathfrak{L}_{j,i}^{\frac{\beta}{2}}\circ\boldsymbol{\Gamma}_{1,i}^{\frac{1}{2}}\mathbf{x}_i\right\|_2^2 \\ &\leq \sum_{j=1}^{M}\Gamma_{2,j}\sum_{i=1}^{N}\Gamma_{1,i}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{2-\beta}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{\beta}\left\|\mathbf{x}_i\right\|_2^2. \end{aligned} \tag{4.21}$$

By applying the inequality $\sum_{j=1}^{M}x_j^2 \leq (\sum_{j=1}^{M}x_j)^2$ for any positive real numbers $x_1,\ldots,x_M$ and the Cauchy-Schwarz inequality to the right hand side of Eq. (4.21), we obtain

$$\begin{aligned}\text{Eq. (4.21)} &\leq \sum_{j=1}^{M}\Gamma_{2,j}\left(\sum_{i=1}^{N}\sqrt{\Gamma_{1,i}}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{1-\frac{\beta}{2}}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{\frac{\beta}{2}}\left\|\mathbf{x}_i\right\|_2\right)^2 \\ &\leq \sum_{j=1}^{M}\Gamma_{2,j}\left(\sum_{i=1}^{N}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{\beta}\right)\left(\sum_{i=1}^{N}\Gamma_{1,i}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{2-\beta}\left\|\mathbf{x}_i\right\|_2^2\right). \end{aligned} \tag{4.22}$$

Then, from the definitions of $\Gamma_{2,j}$ and $\Gamma_{1,i}$ in (4.10), we have $\Gamma_{2,j}\sum_{i=1}^{N}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{\beta} \leq 1$ for any $j=1,\ldots,M$ and $\Gamma_{1,i}\sum_{j=1}^{M}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{2-\beta} \leq 1$ for any $i=1,\ldots,N$, which yields

$$\begin{aligned}\text{Eq. (4.22)} &\leq \sum_{j=1}^{M}\sum_{i=1}^{N}\Gamma_{1,i}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{2-\beta}\left\|\mathbf{x}_i\right\|_2^2 \\ &= \sum_{i=1}^{N}\Gamma_{1,i}\left(\sum_{j=1}^{M}\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}^{2-\beta}\right)\left\|\mathbf{x}_i\right\|_2^2 \\ &\leq \sum_{i=1}^{N}\left\|\mathbf{x}_i\right\|_2^2 = \left\|\mathbf{x}\right\|_2^2. \end{aligned} \tag{4.23}$$

Therefore, we finally obtain

$$\left\|\boldsymbol{\Gamma}_2^{\frac{1}{2}}\circ\mathfrak{L}\circ\boldsymbol{\Gamma}_1^{\frac{1}{2}}\right\|_{\mathrm{op}}^2 = \sup_{\mathbf{x}\neq\mathbf{0}}\frac{\|\boldsymbol{\Gamma}_2^{\frac{1}{2}}\circ\mathfrak{L}\circ\boldsymbol{\Gamma}_1^{\frac{1}{2}}\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} \leq \frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} = 1.$$

□

Remark 4.3.3. To guarantee the convergence of P-PDS, inequality (2.32) has to be strict, but inequality (4.16) is not. However, we do not observe any convergence issue of P-PDS with our preconditioners in the experiments (see Section IV). This is because, our method separates $\mathfrak{L}$ variable by variable and sums up the upper bounds of the operator norms, resulting in setting preconditioners such that $\|\boldsymbol{\Gamma}_2^{\frac{1}{2}}\circ\mathfrak{L}\circ\boldsymbol{\Gamma}_1^{\frac{1}{2}}\|_{\mathrm{op}}^2 < 1$ in almost all real-world applications.

Theorem 4.3.2 asserts that our preconditioners defined in (4.12), (4.13), and (4.14) satisfy the convergence condition of P-PDS in (2.32). Therefore, P-PDS with OVDP generates sequences that converge to an optimal solution of Prob. (4.1).

Here, each $\mu_{j,i}$ is determined in the following manner.

- If the operator norm $\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}$ is known, we set $\mu_{j,i}$ to $\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}$.
- If $\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}$ is unknown, we set $\mu_{j,i}$ to some known or computable upper bound of $\left\|\mathfrak{L}_{j,i}\right\|_{\mathrm{op}}$.

Table 4.1: Features of Existing Methods and Our Method (Highlighted in Bold).

Methods	Parameters requiring manual adjustment	Maintaining proximability	Avoiding access to representation matrices
SP [32]	γ_1	✓	✓
ASP [148]	None.	×	×
PDP [118]	τ	×	✓
OVDP1	None.	✓	✓
OVDP2	None.	✓	✓
OVDP3	None.	✓	✓

Table 4.2: Stopping Criteria.

Applications	Stopping criteria
Mixed noise removal	RMSE < 0.005
Unmixing	RMSE < 0.01
Graph signal recovery	RMSE < 0.001

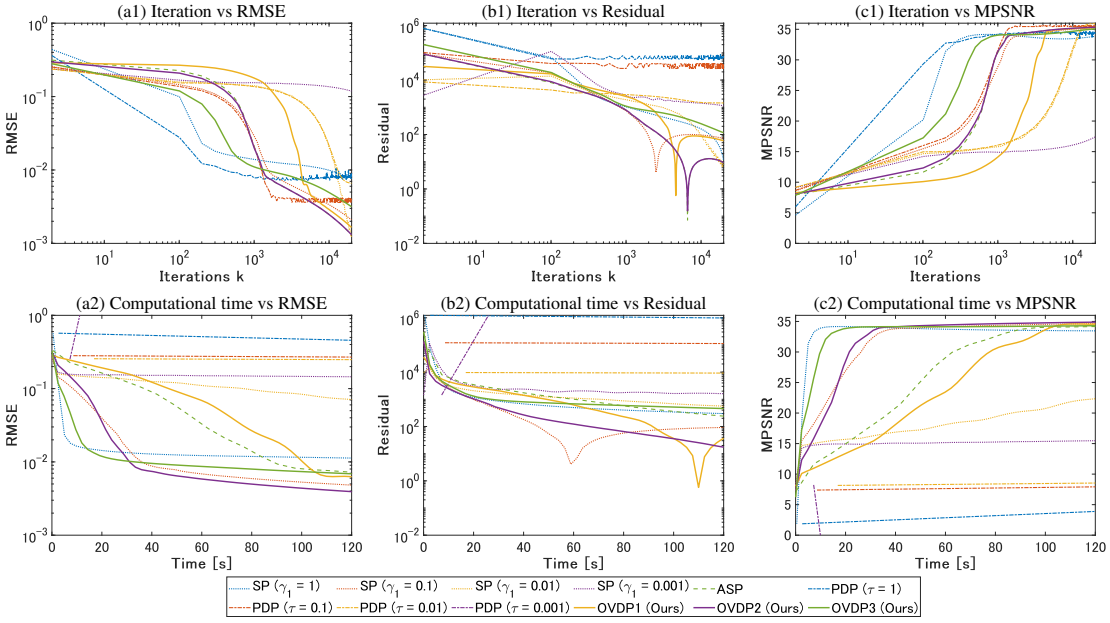


Figure 4.1: Convergence profiles of the mixed noise removal experiments. (a): Iterations/computational time versus RMSE. (b): Iterations/computational time versus Residual. (c): Iterations/computational time versus MPSNR. Note that applying P-PDS with ASP (green dotted line) to Prob. (4.27) is not practical in terms of implementation (the linear operators $\mathfrak{D}_v$, $\mathfrak{D}_h$, and $\mathfrak{D}_b$ are not usually implemented as explicit matrices).

- If the linear operator is the composition of two linear operators $\mathfrak{A}$ and $\mathfrak{B}$ whose operator norms (or their upper bounds) are known ($\|\mathfrak{A}\|_{\mathrm{op}} \leq \alpha_{\mathfrak{A}}, \|\mathfrak{B}\|_{\mathrm{op}} \leq \alpha_{\mathfrak{B}}$), we set $\mu_{j,i}$ to $\alpha_{\mathfrak{A}}\alpha_{\mathfrak{B}}$, which is an upper bound of $\|\mathfrak{A} \circ \mathfrak{B}\|_{\mathrm{op}}$ due to the submultiplicity in (2.11).

Finally, we show the detailed procedures of P-PDS with OVDP in Algorithm 2.

4.4 Experiments and Discussion

In this section, we apply our OVDP to three signal estimation problems: mixed noise removal of hyperspectral images, hyperspectral unmixing, and graph signal recovery. Through these applications, we illustrate the effectiveness and usefulness of our method as follows:

- P-PDS with OVDP is fast on average to obtain an optimal solution of the target optimization problem.
- The preconditioners by OVDP can be easily calculated by using operator norms even if the target optimization problem involves linear operators implemented not as explicit matrices.
- P-PDS with OVDP is efficiently computed by avoiding the computations of skewed proximity operators.

4.4.1 Experimental Setup

We compared OVDP with three existing preconditioner design methods (see Tab. 4.1): the Scalar Preconditioning (SP) [32] in (4.2), the row/column Absolute Sum-based element-wise Preconditioning (ASP) [148] in (4.5), and the Positive-Definite Preconditioning (PDP) [118] in (4.7) and in (4.8). Note that the preconditioners by SP and PDP have parameters (γ_1, τ, θ) to be adjusted manually. For SP, we set γ_1 and γ_2 in (4.2) as $\gamma_1 = 1, 0.1, 0.01, 0.001$, and as in (4.4). The parameter τ in (4.7) and in (4.8) was set as $\tau = 1, 0.1, 0.01, 0.001$. The parameter θ in (4.7) and in (4.8) was set as $\theta = 0.01$, which is recommended in [118]. To calculate skewed proximity operators in the iterations of P-PDSs with ASP and PDP, we used the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [13] initialized with a zero vector.

To check the convergence of P-PDS, we used the Root Mean Square Error (RMSE):

$$\mathrm{RMSE}(\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)}) := \sqrt{\frac{\sum_{i=1}^N \|\mathbf{x}_i^{(t)} - \mathbf{x}_i^*\|_2^2}{\sum_{i=1}^N n_i}}, \tag{4.24}$$

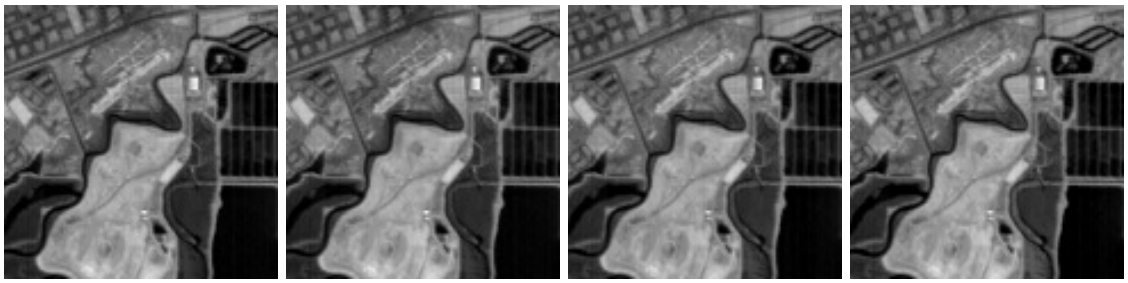
and the residual of the function values:

$$\begin{aligned} &\mathrm{Residual}(\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)}) \\ &:= \left| \left(\sum_{i=1}^N f_i(\mathbf{x}_i^{(t)}) + \sum_{j=1}^M g_j \left(\sum_{i=1}^N \mathfrak{L}_{j,i}(\mathbf{x}_i^{(t)}) \right) \right) - \left(\sum_{i=1}^N f_i(\mathbf{x}_i^*) + \sum_{j=1}^M g_j \left(\sum_{i=1}^N \mathfrak{L}_{j,i}(\mathbf{x}_i^*) \right) \right) \right|, \end{aligned} \tag{4.25}$$

where $\mathbf{x}_1^*, \ldots, \mathbf{x}_N^*$ are oracle solutions. However, such oracle solutions are not available in the experiments, and therefore, we generated pseudo-oracle solutions by the following procedures. We calculated the results through $100,000$ iterations of P-PDS with all the methods in advance, and then selected the best ones among them.

Tab. 4.2 shows the stopping criteria with RMSE as the threshold used in the experiments. Since convergence speeds are different depending on problems, reasonable criteria are also different. To determine reasonable criteria, we employed normalized error ($\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2 / \|\mathbf{x}^{(t)}\|_2$), which is often used as stopping criteria in real-world applications. Based on the normalized error, we set the stopping criteria as the RMSE values such that $\|\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)}\|_2 / \|\mathbf{x}^{(t)}\|_2 < 10^{-5}$.

(a) MPSNR [dB] (b) 14.37 [dB] (c) 34.60 [dB] (d) 34.62 [dB]



(e) 35.45 [dB] (f) 34.81 [dB] (g) 34.66 [dB] (h) 34.62 [dB]

Figure 4.2: Mixed noise removal results. (a): The ground truth HS image. (b): The observed HS image. (c): The HS image estimated by P-PDS with SP [32] ($\gamma_1 = 0.1$). (d): The HS image estimated by P-PDS with ASP [148]. (e): The HS image estimated by P-PDS with PDP [118] ($\tau = 0.1$). (f): The HS image estimated by P-PDS with OVDP1 (Ours). (g): The HS image estimated by P-PDS with OVDP2 (Ours). (h): The HS image estimated by P-PDS with OVDP3 (Ours).

4.4.2 Application to Mixed Noise Removal of Hyperspectral Images

Hyperspectral (HS) images often suffer from various noises, such as random noise, outliers, missing values, and stripe noise, due to environmental and sensor issues [210, 217, 218]. These noises seriously degrade the performance of subsequent processing, such as HS unmixing [68], classification [10], and anomaly detection [170]. Therefore, removing mixed noise from HS images is a crucial preprocessing. Popular mixed noise removal methods adopt the Spatio-Spectral Total Variation (SSTV) regularization [8,61,75,85,132,174,184], which models the spatial piecewise smoothness and the spectral correlations of HS images.

Problem Formulation

Consider that an observed HS image (of size $N_1 \times N_2 \times N_3$) $\mathbf{v} \in \mathbb{R}^{N_1 N_2 N_3}$ is given by

$$\mathbf{v} = \bar{\mathbf{u}} + \bar{\mathbf{s}} + \bar{\mathbf{l}} + \mathbf{n}, \tag{4.26}$$

where $\bar{\mathbf{u}}$, $\bar{\mathbf{s}}$, $\bar{\mathbf{l}}$, and $\mathbf{n}$ are the true HS image of interest, sparsely distributed noise (e.g. outliers and missing values), stripe noise, and random noise, respectively. Based on this observation model, the SSTV-regularized mixed noise removal problem is formulated as the following convex optimization problem:

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{s},\mathbf{l}} \|\mathfrak{D}_v(\mathfrak{D}_b(\mathbf{u}))\|_1 + \|\mathfrak{D}_h(\mathfrak{D}_b(\mathbf{u}))\|_1 + \lambda\|\mathbf{l}\|_1 \\ &\text{s.t.} \begin{cases} \mathfrak{D}_v(\mathbf{l}) = \mathbf{0}, \\ \mathbf{s} \in B^{\mathbf{0}}_{1,\boldsymbol{\eta}_\mathbf{s}}, \\ \mathbf{u} + \mathbf{s} + \mathbf{l} \in B^{\mathbf{v}}_{2,\varepsilon}, \end{cases} \end{aligned} \tag{4.27}$$

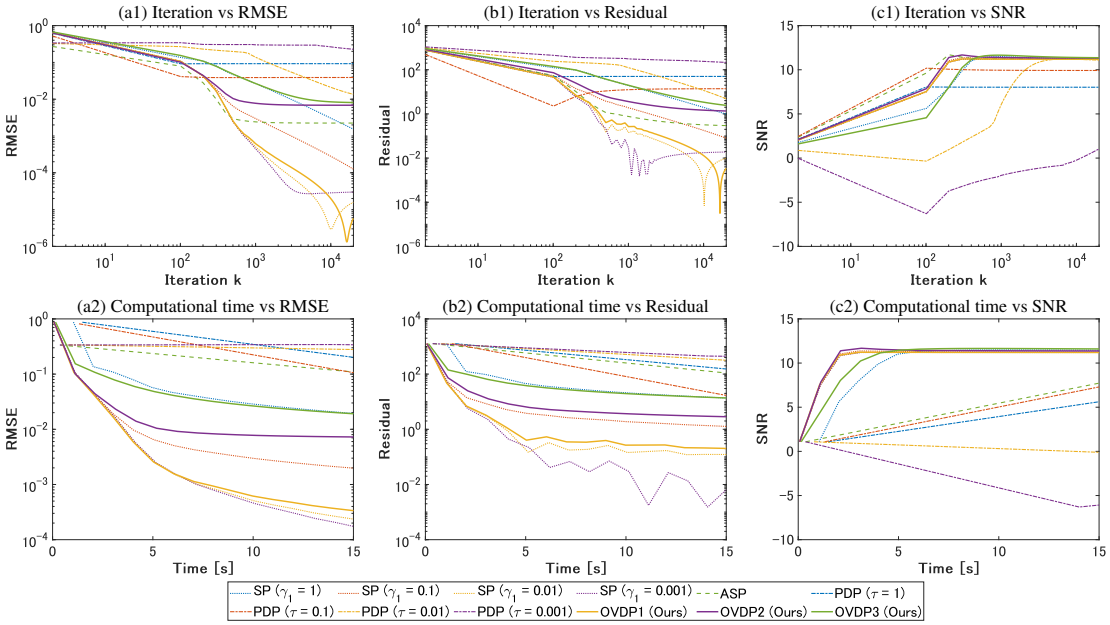


Figure 4.3: Convergence profiles of the unmixing experiments. (a): Iterations/computational time versus RMSE. (b): Iterations/computational time versus Residual. (c): Iterations/computational time versus SNR.

where $\mathfrak{D}_v$, $\mathfrak{D}_h$, and $\mathfrak{D}_b$ are the vertical, horizontal, and spectral difference operators, respectively, with the Neumann boundary condition. To reduce computing resources, these difference operators are usually implemented not as matrices but as the following procedures:

$$[\mathfrak{D}_v(\mathbf{x})]_{i,j,k} := \begin{cases} [\mathbf{x}]_{i,j,k} - [\mathbf{x}]_{i+1,j,k}, & \text{if } i < N_1; \\ 0, & \text{otherwise,} \end{cases} \tag{4.28}$$

$$[\mathfrak{D}_h(\mathbf{x})]_{i,j,k} := \begin{cases} [\mathbf{x}]_{i,j,k} - [\mathbf{x}]_{i,j+1,k}, & \text{if } j < N_2; \\ 0, & \text{otherwise,} \end{cases} \tag{4.29}$$

$$[\mathfrak{D}_b(\mathbf{x})]_{i,j,k} := \begin{cases} [\mathbf{x}]_{i,j,k} - [\mathbf{x}]_{i,j,k+1}, & \text{if } k < N_3; \\ 0, & \text{otherwise,} \end{cases} \tag{4.30}$$

where $[\mathbf{x}]_{i_1,i_2,i_3}$ is the value of $\mathbf{x}$ at a location (i_1, i_2, i_3). Here, $\|\cdot\|_1$ is the ℓ_1 norm, and $B^{\mathbf{v}}_{2,\varepsilon}$ and $B^{\mathbf{0}}_{1,\eta_\mathbf{s}}$ are the ℓ_2 and ℓ_1 norm balls, respectively given by

$$\begin{aligned} B^{\mathbf{v}}_{2,\varepsilon} &:= \left\{ \mathbf{x} \in \mathbb{R}^{N_1 N_2 N_3} \;\middle|\; \|\mathbf{v} - \mathbf{x}\|_2 \le \varepsilon \right\}, \\ B^{\mathbf{0}}_{1,\eta_\mathbf{s}} &:= \left\{ \mathbf{x} \in \mathbb{R}^{N_1 N_2 N_3} \;\middle|\; \|\mathbf{x}\|_1 \le \eta_\mathbf{s} \right\}. \end{aligned} \tag{4.31}$$

The term $\|\mathfrak{D}_v(\mathfrak{D}_b(\mathbf{u}))\|_1 + \|\mathfrak{D}_h(\mathfrak{D}_b(\mathbf{u}))\|_1$ is the SSTV regularization. The positive value λ is a balancing parameter between the SSTV regularization and the sparse noise term. The hard constraint guarantees the ℓ_2 data-fidelity to $\mathbf{v}$ with the radius $\varepsilon \ge 0$.[2]

[2]The original SSTV-regularized denoising formulation proposed in [8] incorporates an ℓ_2 data-fidelity term as a part of the objective function, whereas the formulation in (4.27) imposes data fidelity as an ℓ_2-ball constraint. These two formulations are essentially the same with appropriate hyperparameters, but a constrained formulation like (4.27) is preferred in experimental comparison and real-world applications because it facilitates hyperparameter settings as adopted, e.g., in Refs. [7, 47, 137, 138, 141].

Table 4.3: The Preconditioners by OVDP for Mixed Noise Removal.

	$\boldsymbol{\Gamma}_{1,1}$	$\boldsymbol{\Gamma}_{1,2}$	$\boldsymbol{\Gamma}_{1,3}$	$\boldsymbol{\Gamma}_{2,1}$	$\boldsymbol{\Gamma}_{2,2}$	$\boldsymbol{\Gamma}_{2,3}$	$\boldsymbol{\Gamma}_{2,4}$
OVDP1	$\frac{1}{33}\mathbf{I}$	$\mathbf{I}$	$\frac{1}{5}\mathbf{I}$	$\frac{1}{3}\mathbf{I}$	$\frac{1}{3}\mathbf{I}$	$\frac{1}{3}\mathbf{I}$	$\frac{1}{3}\mathbf{I}$
OVDP2	$\frac{1}{9}\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$	$\frac{1}{4}\mathbf{I}$	$\frac{1}{4}\mathbf{I}$	$\frac{1}{33}\mathbf{I}$	$\frac{1}{3}\mathbf{I}$
OVDP3	$\frac{1}{33}\mathbf{I}$	$\frac{1}{33}\mathbf{I}$	$\frac{1}{33}\mathbf{I}$	$\frac{1}{33}\mathbf{I}$	$\frac{1}{33}\mathbf{I}$	$\frac{1}{33}\mathbf{I}$	$\frac{1}{33}\mathbf{I}$

By using the indicator function (see Eq. (2.13)) of $B^{\mathbf{v}}_{2,\varepsilon}$, Prob. (4.27) is reduced to Prob. (4.1) through the following reformulation:

$$\begin{aligned}\min_{\substack{\mathbf{u},\mathbf{s},\mathbf{l},\\ \mathbf{z}_1,\mathbf{z}_2,\mathbf{z}_3,\mathbf{z}_4}} \quad & \iota_{B^{\mathbf{0}}_{1,\eta_{\mathbf{s}}}}(\mathbf{s}) + \lambda\|\mathbf{l}\|_1 + \|\mathbf{z}_1\|_1 + \|\mathbf{z}_2\|_1 + \iota_{\{\mathbf{0}\}}(\mathbf{z}_3) + \iota_{B^{\mathbf{v}}_{2,\varepsilon}}(\mathbf{z}_4) \\ \text{s.t.} \quad & \begin{cases} \mathbf{z}_1 = \mathfrak{D}_v(\mathfrak{D}_b(\mathbf{u})), \\ \mathbf{z}_2 = \mathfrak{D}_h(\mathfrak{D}_b(\mathbf{u})), \\ \mathbf{z}_3 = \mathfrak{D}_v(\mathbf{l}), \\ \mathbf{z}_4 = \mathbf{u} + \mathbf{s} + \mathbf{l}. \end{cases} \end{aligned} \tag{4.32}$$

Applying Algorithm 2 to Prob. (4.32), we can compute an optimal solution of Prob. (4.27). Here, since it is satisfied that $\|\mathfrak{D}_v \circ \mathfrak{D}_b\|_{\mathrm{op}} \leq 4$, $\|\mathfrak{D}_h \circ \mathfrak{D}_b\|_{\mathrm{op}} \leq 4$,[3] and $\|\mathbf{I}\|_{\mathrm{op}} = 1$, the preconditioners designed by OVDP are given in Tab. 4.3.

Experimental Results and Discussion

For SP, μ_{SP} in (4.4) was set as

$$\mu_{SP} = \sqrt{39}, \tag{4.33}$$

because the following inequality holds due to the inequality of the operator norms of block matrices [18]:

$$\begin{aligned} &\left\| \begin{bmatrix} \mathfrak{D}_v \circ \mathfrak{D}_b & \mathbf{O} & \mathbf{O} \\ \mathfrak{D}_h \circ \mathfrak{D}_b & \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathfrak{D}_v \\ \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix} \right\|^2_{\mathrm{op}} \\ &\leq \|\mathfrak{D}_v \circ \mathfrak{D}_b\|^2_{\mathrm{op}} + \|\mathfrak{D}_h \circ \mathfrak{D}_b\|^2_{\mathrm{op}} + \|\mathfrak{D}_v\|^2_{\mathrm{op}} + 3\,\|\mathbf{I}\|^2_{\mathrm{op}} \\ &< 4^2 + 4^2 + 2^2 + 3 \times 1^2 = 39, \end{aligned} \tag{4.34}$$

where $\mathbf{O}$ is a zero operator.

We also derived the preconditioners in (4.5), for (4.32). Let us remark that since $\mathfrak{D}_v$, $\mathfrak{D}_h$, and $\mathfrak{D}_b$ in (4.32) are not usually implemented as explicit matrices, applying ASP to (4.32) is not practical in real-world applications. Let $\mathbf{x} \in \mathbb{R}^{n_1 n_2 n_3}$ be a vectorized data cube and $[\mathbf{x}]_{i_1,i_2,i_3}$ be the value of $\mathbf{x}$ at a location (i_1, i_2, i_3). Then the preconditioners are

$$\begin{aligned} &\boldsymbol{\Gamma}_{1,1} = \mathrm{diag}(\mathbf{g}_1), \boldsymbol{\Gamma}_{1,2} = \mathbf{I}, \boldsymbol{\Gamma}_{1,3} = \mathrm{diag}(\mathbf{g}_2), \\ &\boldsymbol{\Gamma}_{2,1} = \boldsymbol{\Gamma}_{2,2} = \frac{1}{4}, \boldsymbol{\Gamma}_{2,3} = \frac{1}{2}, \boldsymbol{\Gamma}_{2,4} = \frac{1}{3}\mathbf{I}. \end{aligned} \tag{4.35}$$

[3]These are derived from $\|\mathfrak{D}_v\|_{\mathrm{op}} \leq 2$, $\|\mathfrak{D}_v\|_{\mathrm{op}} \leq 2$, $\|\mathfrak{D}_v\|_{\mathrm{op}} \leq 2$ [30], and the submultiplicity of operator norms (Eq. (2.11))

Here, $\mathbf{g}_1 \in \mathbb{R}^{N_1N_2N_3}$ and $\mathbf{g}_2 \in \mathbb{R}^{N_1N_2N_3}$ are given as follows:

$$[\mathbf{g}_1]_{i_1,i_2,i_3} = \begin{cases} \frac{1}{9}, & \text{if } i_1 \in I_1 \text{ and } i_2 \in I_2 \text{ and } i_3 \in I_3; \\ \frac{1}{3}, & \text{if } i_1 \in E_1 \text{ and } i_2 \in E_2 \text{ and } i_3 \in E_3; \\ \frac{1}{4}, & \text{if } i_3 \in E_3 \text{ and } ((i_1 \in E_1 \text{ and } i_2 \in I_2) \text{ or } (i_1 \in I_1 \text{ and } i_2 \in E_2)); \\ \frac{1}{5}, & \text{if } i_1 \in E_1 \text{ and } i_2 \in E_2 \text{ and } i_3 \in I_3; \\ \frac{1}{7}, & \text{otherwise,} \end{cases} \tag{4.36}$$

$$[\mathbf{g}_2]_{i_1,i_2,i_3} = \begin{cases} \frac{1}{3}, & \text{if } i_1 \in I_1; \\ \frac{1}{2}, & \text{otherwise,} \end{cases} \tag{4.37}$$

where I_m and E_m for $m = 1, 2, 3$ are $\{2, \ldots, n_m - 1\}$ and $\{1, n_m\}$, respectively. In this case, the skewed proximity operators are separable and thus have analytical solutions. This indicates that P-PDS with ASP does not require FISTA.

As the ground truth HS image, we used Moffett Field [1] of size $120 \times 120 \times 176$. The observed image was generated by adding white Gaussian noise with the standard deviation $\sigma = 0.05$ and Salt & Pepper noise with the ratio $p_{\mathbf{s}} = 0.1$. The parameters λ, $\eta_{\mathbf{s}}$, and ε were set to 0.005, $0.5 * 0.95 * p_{\mathbf{s}} * N_1N_2N_3$, and $0.95\sigma\sqrt{(1-p_{\mathbf{s}})N_1N_2N_3}$, respectively. For the quantitative evaluation of image qualities, we used the Mean Peak Signal-to-Noise Ratio (MPSNR):

$$\mathrm{MPSNR}(\mathbf{u}^{(t)}) := \frac{1}{N_3}\sum_{b=1}^{N_3} 10\log_{10}\left(\frac{N_1N_2}{\|\bar{\mathbf{u}}_b - \mathbf{u}_b^{(t)}\|_2^2}\right), \tag{4.38}$$

where $\bar{\mathbf{u}}_b$ and $\mathbf{u}_b^{(t)}$ are the bth band of the ground-truth image $\bar{\mathbf{u}}$ and the estimated image $\mathbf{u}^{(t)}$.

Figure 4.1 plots iterations versus RMSE, Residual, and MPSNR and computational time versus RMSE, Residual, and MPSNR, respectively. In terms of iterations (Figs. 4.1 (a1), (b1), and (c1)), P-PDSs with SP ($\gamma_1 = 0.01$), SP ($\gamma_1 = 0.001$), PDP ($\tau = 0.01$), and PDP ($\tau = 0.001$) were very slow, and P-PDSs with SP ($\gamma_1 = 1$), SP ($\gamma_1 = 0.1$), ASP, PDP ($\tau = 1$), PDP ($\tau = 0.1$), OVDP2, and OVDP3 were fast. For P-PDS with OVDP1 , the evolution of the MPSNR values was slightly slow, but the convergence of the RMSE and Residual values was not. In terms of computational time (Figs. 4.1 (a2), (b2), and (c2)), although P-PDSs with SP, ASP, and OVDP have the same computational complexity per iteration in O-notation, P-PDS with ASP took longer than P-PDSs with SP and OVDP. When computing the analytic solutions of the proximity operators, P-PDSs with SP and OVDP require the multiplication of a scalar and a vector, while P-PDS with ASP requires the element-wise multiplication of two vectors. Since the latter takes longer to run than the former, P-PDS with ASP was longer in running time. P-PDSs with PDP were very slow because they require the iterative algorithm to calculate the skewed proximity operator.

Figure 4.2 shows the denoising results and the MPSNR values [dB] obtained by P-PDS with SP ($\gamma_1 = 0.1$), ASP, PDP ($\tau = 0.1$), OVDP1, OVDP2, and OVDP3. The algorithm was run until satisfying the stopping criterion or reaching 10000 iterations. We can see that all results are almost the same in terms of the MPSNR and the visual qualities.

4.4.3 Application to Hyperspectral Unmixing

An HS image is a three-dimensional data cube that consists of two-dimensional spatial information and one-dimensional spectral information. Compared to grayscale or RGB images, HS images offer more than several hundred bands, each of which contains specific unique wavelength characteristics of materials such as minerals, soils, and liquids. Due to the trade-off between spatial resolution and wavelength resolution, HS sensors do not have

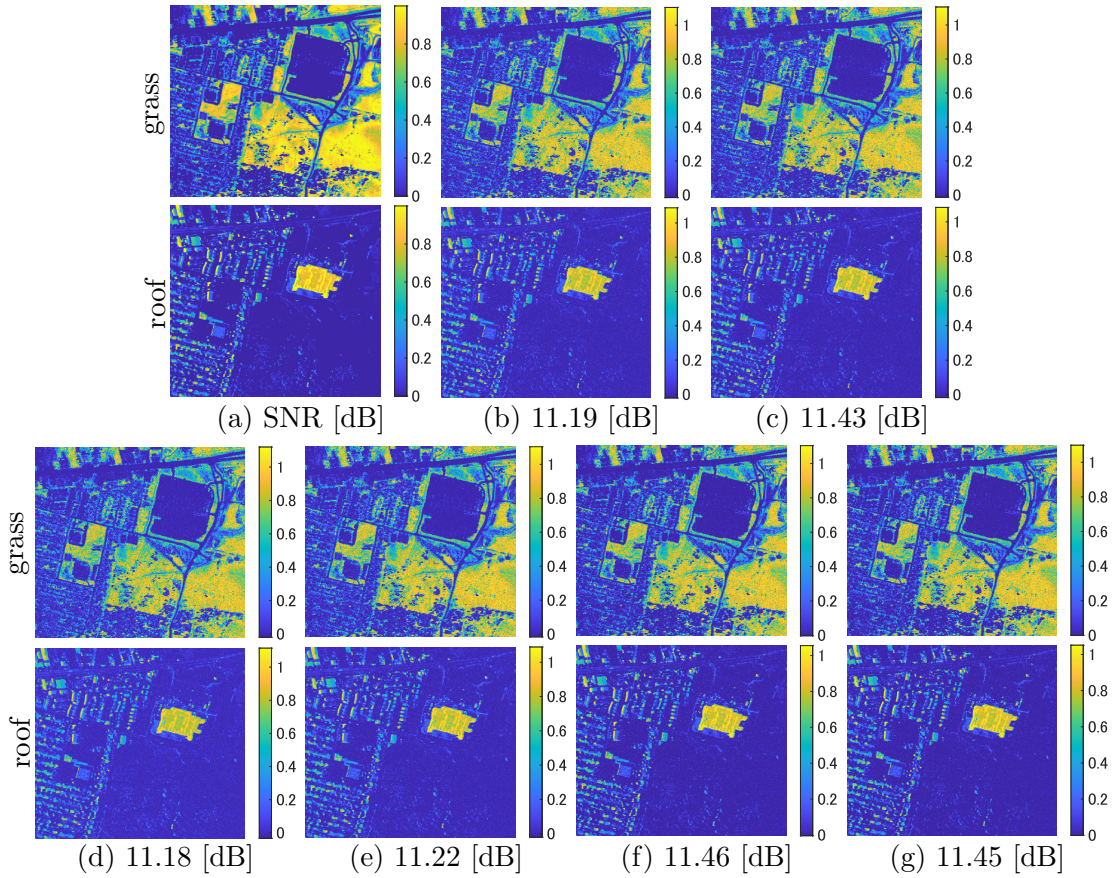


Figure 4.4: Abundance maps of HS unmixing results. (a): The ground truth abundance maps. (b): The abundance maps estimated by P-PDS with SP [32] ($\gamma_1 = 0.001$). (c): The abundance maps estimated by P-PDS with ASP [148]. (d): The abundance maps estimated by P-PDS with PDP [118] ($\tau = 0.01$). (e): The abundance maps estimated by P-PDS with OVDP1 (Ours). (f): The abundance maps estimated by P-PDS with OVDP2 (Ours). (g): The abundance maps estimated by P-PDS with OVDP3 (Ours).

a sufficient spatial resolution, resulting in containing multiple components (called endmembers) in a pixel [95], which is referred to as a mixel. The process of decomposing the mixels into endmembers and their abundance maps is called unmixing. Unmixing has been actively studied in the remote sensing field because of its indispensability for analyzing HS images [68, 124]. One of the popular unmixing methods is the constrained collaborative sparse regression problem [87], which has attracted attention as an optimization-based strategy for HS unmixing [9, 183, 205].

Problem Formulation

Let $\mathbf{v}_i \in \mathbb{R}^{N_3 \times 1}$ represent an N_3-dimensional ith pixel vector of an HS image with N_3 spectral bands and $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_m] \in \mathbb{R}^{N_3 \times m}$ be an endmember matrix that denotes a spectral library with m spectral signatures. The pixel $\mathbf{v}_i$ can be modeled as the following form of linear combination:

$$\mathbf{v}_i = \mathbf{E}\mathbf{a}_i + \mathbf{n}_i, \tag{4.39}$$

where $\mathbf{a}_i \in \mathbb{R}^{N_3 \times 1}$ is an abundance map. Introducing the extended endmember matrix $\widetilde{\mathbf{E}} = \mathrm{diag}(\mathbf{E}, \ldots, \mathbf{E}) \in \mathbb{R}^{N_1 N_2 N_3 \times N_1 N_2 m}$, we can express an observed HS image $\mathbf{v} =$

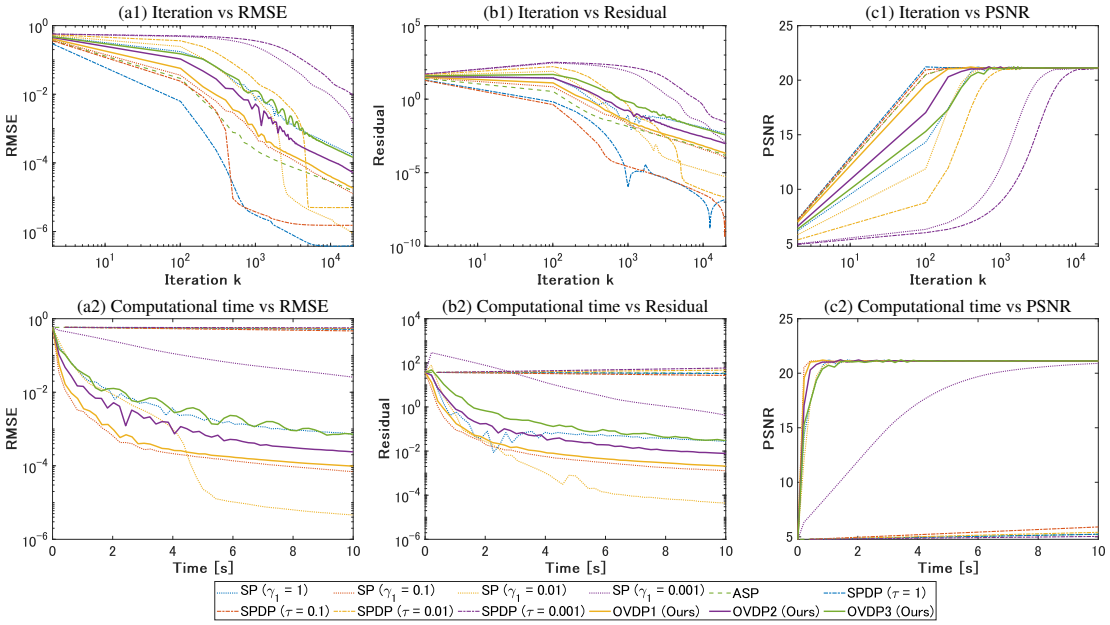


Figure 4.5: Convergence profiles of the graph singal recovery experiments. (a): Iterations/computational time versus RMSE. (b): Iterations/computational time versus Residual. (c): Iterations/computational time versus PSNR.

$[\mathbf{v}_1^\top, \ldots, \mathbf{v}_{N_1N_2}^\top]^\top$ as

$$\mathbf{v} = \widetilde{\mathbf{E}}\mathbf{a} + \mathbf{n}. \tag{4.40}$$

Based on the above model, the constrained collaborative sparse regression problem of unmixing is formulated as the following convex optimization problem:

$$\min_{\mathbf{a}} \|\mathbf{a}\|_{1,2} \text{ s.t. } \begin{cases} \widetilde{\mathbf{E}}\mathbf{a} \in B^{\mathbf{v}}_{2,\varepsilon}, \\ \mathbf{a} \in \mathbb{R}_+^{N_1N_2N_3}. \end{cases} \tag{4.41}$$

The first term is the mixed $\ell_{1,2}$ norm, which is defined by

$$\|\mathbf{a}\|_{1,2} = \sum_{e=1}^{m} \sqrt{\sum_{i=1}^{N_1N_2} [\mathbf{a}_i]_e^2}. \tag{4.42}$$

The first constraint serves as data-fidelity with the $\mathbf{v}$-centered ℓ_2-ball of the radius $\varepsilon > 0$.[4] The second constraint enforces $\mathbf{a}$ to belong to the nonnegative orthant $\mathbb{R}_+^{N_1N_2N_3}$.

By using the indicator functions (see Eq. (2.13)) of $B^{\mathbf{v}}_{2,\varepsilon}$ and $\mathbb{R}_+^{N_1N_2N_3}$, Prob. (4.41) is reduced to Prob. (4.1) via the following reformulation:

$$\begin{aligned} &\min_{\mathbf{a},\mathbf{z}_1,\mathbf{z}_2} \|\mathbf{a}\|_{1,2} + \iota_{B^{\mathbf{v}}_{2,\varepsilon}}(\mathbf{z}_1) + \iota_{\mathbb{R}_+^{N_1N_2N_3}}(\mathbf{z}_2) \\ &\text{s.t.} \begin{cases} \mathbf{z}_1 = \widetilde{\mathbf{E}}\mathbf{a}, \\ \mathbf{z}_2 = \mathbf{a}. \end{cases} \end{aligned} \tag{4.43}$$

Applying Algorithm 2 to Prob. (4.43), we can obtain an optimal solution of Prob. (4.41). Since the functions $\|\cdot\|_{1,2}$ and $\iota_{B^{\mathbf{v}}_{2,\varepsilon}}$ are not separable for each element of the input variable, an iterative algorithm is needed for the calculation of their skewed proximity operators relative to the metric induced by the preconditioners of ASP and PDP. Here, the preconditioners designed by OVDP are as in Tab. 4.4.

[4]The original constrained collaborative sparse regression formulation proposed in [87] incorporates an ℓ_2 data-fidelity term as a part of the objective function, whereas the formulation in (4.41) imposes data fidelity as an ℓ_2-ball constraint. The reason is similar to the case of the mixed noise removal experiment.

Table 4.4: The Preconditioners by OVDP for Unmixing.

	$\boldsymbol{\Gamma}_{1,1}$	$\boldsymbol{\Gamma}_{2,1}$	$\boldsymbol{\Gamma}_{2,2}$
OVDP1	$\frac{1}{\|\widetilde{\mathbf{E}}\|_{\mathrm{op}}^2+1^2}\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$
OVDP2	$\frac{1}{\|\widetilde{\mathbf{E}}\|_{\mathrm{op}}+1}\mathbf{I}$	$\frac{1}{\|\widetilde{\mathbf{E}}\|_{\mathrm{op}}}\mathbf{I}$	$\mathbf{I}$
OVDP3	$\frac{1}{2}\mathbf{I}$	$\frac{1}{\|\widetilde{\mathbf{E}}\|_{\mathrm{op}}^2}\mathbf{I}$	$\mathbf{I}$

Experimental Results and Discussion

For SP, μ_{SP} in (4.4) was set as

$$\mu_{SP} = \sqrt{\|\widetilde{\mathbf{E}}\|_{\mathrm{op}}^2 + 1}, \tag{4.44}$$

because the following inequality holds due to the inequality of the operator norms of block matrices [18]:

$$\left\| \begin{bmatrix} \widetilde{\mathbf{E}} \\ \mathbf{I} \end{bmatrix} \right\|_{\mathrm{op}}^2 \leq \|\widetilde{\mathbf{E}}\|_{\mathrm{op}}^2 + \|\mathbf{I}\|_{\mathrm{op}}^2 = \|\widetilde{\mathbf{E}}\|_{\mathrm{op}}^2 + 1. \tag{4.45}$$

For PDP, the preconditioners in (4.8) were used since the number of dual variables is two.

As the ground truth HS image, we used the urban dataset[5], which has been widely used in the field of HS unmixing. The image consists of 307×307 pixels with 210 spectral bands. In the image, six main endmembers can be observed in the scene: asphalt road, grass, tree, roof, metal, and dirt. The observed data was generated by adding white Gaussian noise with the standard deviation $\sigma = 0.05$. The parameter ε was set to $0.9\sigma\sqrt{N_1N_2N_3}$. For the quantitative evaluation of image qualities, we used the Signal-to-Noise Ratio (SNR) [6]:

$$\mathrm{SNR}(\mathbf{a}^{(t)}) := 10\log_{10}\left(\frac{\|\bar{\mathbf{a}}\|_2}{\|\mathbf{a}^{(t)} - \bar{\mathbf{a}}\|_2}\right), \tag{4.46}$$

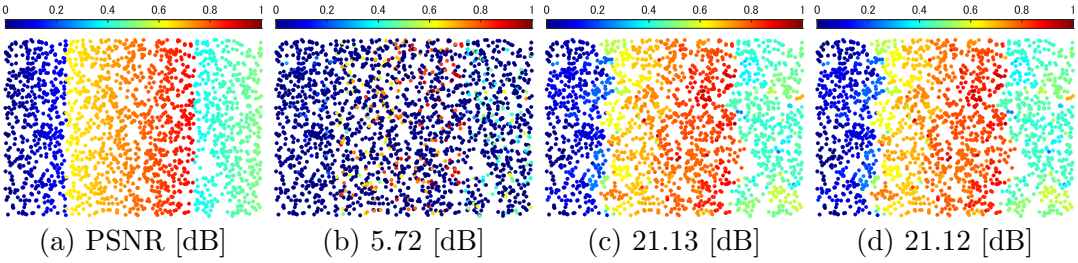
where $\mathbf{a}^{(t)}$ and $\bar{\mathbf{a}}$ are the estimated and ground true abundance maps, respectively.

Figure 4.3 plots iterations versus RMSE, Residual, and SNR and computational time versus RMSE, Residual, and SNR, respectively. In terms of iterations (Figs. 4.3 (a1), (b1), and (c1)), P-PDS with PDP was very slow in all parameter cases. P-PDSs with SP ($\gamma_1 = 1$), OVDP2, and OVDP3 were slightly slow, but P-PDSs with SP ($\gamma_1 = 0.1$) and ASP were not. P-PDSs with SP ($\gamma_1 = 0.01$), SP ($\gamma_1 = 0.001$), and OVDP1 were fast. In terms of computational time (Figs. 4.3 (a2), (b2), and (c2)), P-PDS with SP and OVDP were similar to the results with respect to iterations. P-PDSs with ASP and PDP were very slow because they require the iterative algorithm to calculate the skewed proximity operator in each iteration of P-PDS. At first glance, the curves generated by P-PDSs with PDP ($\tau = 1$, 0.1, and 0.001) may appear to converge to different SNRs. This is because they take enormous amounts of time to converge (in fact, the convergence times are too enormous to measure). Therefore, they do not converge to different SNRs.

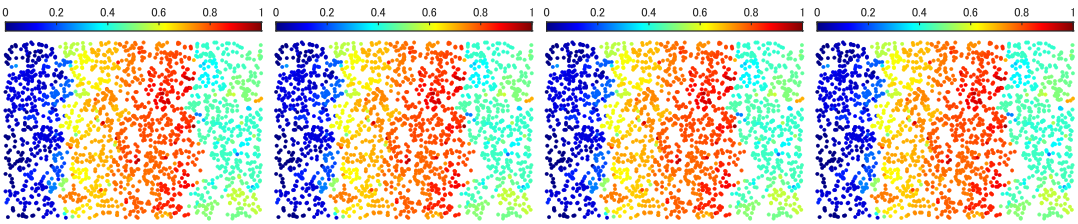
Figure 4.4 shows the unmixing results and the SNR values [dB] obtained by P-PDS with SP ($\gamma_1 = 0.001$), ASP, PDP ($\tau = 0.01$), OVDP1, OVDP2, and OVDP3. The algorithm was run until satisfying the stopping criterion or reaching 10000 iterations. We can see that all results are almost the same in terms of the SNR and the visual qualities.

[5] http://www.tec.army.mil/Hypercube

[6] This evaluation metric is often referred to as the signal to reconstruction error in the literature of HS unmixing (e.g., [87, 183, 205]).



(a) PSNR [dB] (b) 5.72 [dB] (c) 21.13 [dB] (d) 21.12 [dB]



(e) 21.12 [dB] (f) 21.14 [dB] (g) 21.12 [dB] (h) 21.12 [dB]

Figure 4.6: Graph signal recovery results. (a): The ground truth signal. (b): The observed graph signal. (c): The graph signal estimated by P-PDS with SP [32] ($\gamma_1 = 0.1$). (d): The graph signal estimated by P-PDS with ASP [148]. (e): The graph signal estimated by P-PDS with PDP [118] ($\tau = 1$). (f): The graph signal estimated by P-PDS with OVDP1 (Ours). (g): The graph signal estimated by P-PDS with OVDP2 (Ours). (h): The graph signal estimated by P-PDS with OVDP3 (Ours).

4.4.4 Application to Graph Signal Recovery

Graphs explicitly represent the irregular structures of data [143, 159, 165], such as traffic and sensor network data, geographical data, mesh data, and biomedical data. The signals on the irregular structures are called graph signals. Similar to classical signal processing, sampling of graph signals [178] is a leading research topic due to its numerous promising applications, for example, sensor placement, filter bank designs, traffic monitoring, and semi-supervised learning. In graph signal recovery, which reconstructs original graph signals from sampled graph signals, it is assumed that graph signals have some properties, such as smoothness. The smoothness of graph signals can be captured by graph total variation type regularizations [17, 69, 142], which have been applied to various graph signal processing tasks [104, 109].

Problem Formulation

We consider signals on weighted directed graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with a vertex set $\mathcal{V} = \{1, \ldots, N_{\mathcal{G}}\}$, an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and a weighted matrix $\mathbf{W} \in \mathbb{R}^{N_{\mathcal{G}} \times N_{\mathcal{G}}}$. The value $W_{i,j}$ is designed to be large if the relation between vertices i and j is strong. Graph signals are typically assumed to be smooth with respect to the graph $\mathcal{G}$. Based on the assumption, graph signal recovery methods often adopt the graph total variation (GTV) [17, 165]:

$$\|\mathbf{x}\|_{\mathrm{GTV}} := \|\mathbf{D}_{\mathcal{G}}\mathbf{x}\|_{1,2} = \sum_{i=1}^{N_{\mathcal{G}}} \|\mathbf{y}_i\|_2, \tag{4.47}$$

where $\mathbf{D}_{\mathcal{G}}$ is the graph difference operator defined as follows. Let $\mathbf{D}_{\mathcal{G}}\mathbf{x} = [\mathbf{y}_1^\top, \ldots, \mathbf{y}_{N_{\mathcal{G}}}^\top]$, then each $\mathbf{y}_i$ consists of the weighted differences between the graph signal value x_i at an ith vertex and the graph signal values x_j $(\forall j \in \mathcal{N}(i) := \{k \in \mathcal{V} \mid W_{i,k} \neq 0\})$ at its connected vertices $\mathcal{N}(i)$, i.e.,

$$[\mathbf{y}_i]_j := (x_j - x_i)W_{i,j}, \ (\forall j \in \mathcal{N}(i)). \tag{4.48}$$

Table 4.5: The Preconditioners by OVDP for Graph Signal Recovery.

	$\boldsymbol{\Gamma}_{1,1}$	$\boldsymbol{\Gamma}_{2,1}$	$\boldsymbol{\Gamma}_{2,2}$
OVDP1	$\frac{1}{\|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}^2+1^2}\mathbf{I}$	$\mathbf{I}$	$\mathbf{I}$
OVDP2	$\frac{1}{\|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}+1}\mathbf{I}$	$\frac{1}{\|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}}\mathbf{I}$	$\mathbf{I}$
OVDP3	$\frac{1}{2}\mathbf{I}$	$\frac{1}{\|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}^2}\mathbf{I}$	$\mathbf{I}$

By weighting the difference between x_i and x_j by $W_{i,j}$, GTV can capture the graph signal smoothness that the difference of graph signal values is small as the relation of their vertices is strong.

Consider that an observed graph signal $\mathbf{v} \in \mathbb{R}^{M_{\mathcal{G}}}$ is modeled by

$$\mathbf{v} = \boldsymbol{\Phi}\bar{\mathbf{u}} + \mathbf{n}, \tag{4.49}$$

where $\bar{\mathbf{u}} \in \mathbb{R}^{N_{\mathcal{G}}}$, $\mathbf{n} \in \mathbb{R}^{M_{\mathcal{G}}}$, and $\boldsymbol{\Phi} \in \{0,1\}^{M_{\mathcal{G}} \times N_{\mathcal{G}}}$ are the true graph signal of interest, random additive noise, and the sampling matrix, respectively. Based on this observation model, the GTV regularized graph signal recovery problem is formulated as the following convex optimization problem [17]:

$$\min_{\mathbf{u}} \|\mathbf{D}_{\mathcal{G}}\mathbf{u}\|_{\mathrm{GTV}} \text{ s.t. } \boldsymbol{\Phi}\mathbf{u} \in B_{2,\varepsilon}^{\mathbf{v}}. \tag{4.50}$$

The hard constraint guarantees the ℓ_2 data fidelity to the observed signal $\mathbf{v}$ with the radius ε.

By using the indicator function (see Eq. (2.13)) of $B_{2,\varepsilon}^{\mathbf{v}}$, Prob. (4.50) is reduced to Prob. (4.1) via the following reformulation:

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{z}_1,\mathbf{z}_2} \|\mathbf{z}_1\|_{1,2} + \iota_{B_{2,\varepsilon}^{\mathbf{v}}}(\mathbf{z}_2) \\ &\text{s.t. } \begin{cases} \mathbf{z}_1 = \mathbf{D}_{\mathcal{G}}\mathbf{u}, \\ \mathbf{z}_2 = \boldsymbol{\Phi}\mathbf{u}. \end{cases} \end{aligned} \tag{4.51}$$

Applying Algorithm 2 to Prob. (4.51), we can compute an optimal solution of Prob. (4.50). Since the function $\|\cdot\|_{1,2}$ is not separable for each element of the input variable, an iterative algorithm is needed for the computation of their skewed proximity operators relative to the metric induced by the preconditioners of ASP and PDP in (4.5). Here, the preconditioners designed by OVDP are given as in Tab. 4.5. According to [17], an upper bound of the operator norm $\|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}$ can be derived by

$$\|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}} \leq 2 \max_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} (W_{i,j}^2 + W_{j,i}^2). \tag{4.52}$$

An upper bound of the norm of the sampling matrix is one, i.e., $\|\boldsymbol{\Phi}\|_{\mathrm{op}} = 1$.

Experimental Results

For SP, μ_{SP} in (4.4) was set as

$$\mu_{SP} = \sqrt{\|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}^2 + 1}, \tag{4.53}$$

because the following inequality holds due to the inequality of the operator norms of block matrices [18]:

$$\left\| \begin{bmatrix} \mathbf{D}_{\mathcal{G}} \\ \boldsymbol{\Phi} \end{bmatrix} \right\|_{\mathrm{op}}^2 \leq \|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}^2 + \|\boldsymbol{\Phi}\|_{\mathrm{op}}^2 \leq \|\mathbf{D}_{\mathcal{G}}\|_{\mathrm{op}}^2 + 1. \tag{4.54}$$

The preconditioners by ASP in (4.5) for Prob. (4.50) are

$$
\begin{aligned}
[\boldsymbol{\Gamma}_{1,1}]_{i,i} &= \frac{1}{\sum_{j=1}^{N_{\mathcal{G}}N_{\mathcal{G}}} |W_{i,j}| + \sum_{k=1}^{M_{\mathcal{G}}} \Phi_{k,i}}, \ (\forall i = 1, \ldots, N_{\mathcal{G}}), \\
[\boldsymbol{\Gamma}_{2,1}]_{i,i} &= \frac{1}{2\sum_{j=1}^{N_{\mathcal{G}}} |W_{j,i}|}, \ (\forall i = 1, \ldots, N_{\mathcal{G}}N_{\mathcal{G}}), \\
[\boldsymbol{\Gamma}_{2,2}]_{i,i} &= 1, \ (\forall i = 1, \ldots, M_{\mathcal{G}}).
\end{aligned} \tag{4.55}
$$

For PDP, the preconditioners in (4.8) were used since the number of dual variables is two.

We constructed a random sensor graph $\mathcal{G}$ by using GSPBox [147], then generated a noiseless piece-wise smooth graph signal on the graph with $N_{\mathcal{G}} = 2000$ vertices. The observed graph signal was obtained by adding white Gaussian noise with 0.1 of the standard deviation σ and by sampling it with 0.2 of the sampling rate ($M_{\mathcal{G}} = 0.2N_{\mathcal{G}}$). The parameter ε was set as $\varepsilon = 0.9\sigma\sqrt{M_{\mathcal{G}}}$. For the quantitative evaluation of recovery qualities, we used the Peak Signal-to-Noise Ratio (PSNR):

$$
\mathrm{PSNR} := 10\log_{10}\left(\frac{N_{\mathcal{G}}}{\|\bar{\mathbf{u}} - \mathbf{u}^{(t)}\|_2^2}\right), \tag{4.56}
$$

Figure 4.5 plots iterations versus RMSE, Residual, and PSNR and computational time versus RMSE, Residual, and PSNR, respectively. In terms of iterations (Figs. 4.5 (a1), (b1), and (c1)), P-PDSs with SP ($\gamma_1 = 0.001$) and PDP ($\tau = 0.001$) were very slow. P-PDSs with SP ($\gamma_1 = 1$), PDP ($\tau = 0.01$), OVDP3 were not slow but not fast. P-PDSs with SP ($\gamma_1 = 0.1$), SP ($\gamma_1 = 0.01$), ASP, PDP ($\tau = 1$), PDP ($\tau = 0.1$), OVDP1, and OVDP2 were fast. In terms of computational time (Figs. 4.5 (a2), (b2), and (c2)), P-PDS with SP and OVDP were similar to the results with respect to iterations. P-PDSs with ASP and PDP were very slow because they require the iterative algorithm to calculate the skewed proximity operator.

Figure 4.6 shows the recovery results and the PSNR values [dB] obtained by P-PDS with SP ($\gamma_1 = 0.1$), ASP, PDP ($\tau = 1$), OVDP1, OVDP2, and OVDP3. The algorithm was run until satisfying the stopping criterion or reaching 10000 iterations. We can see that all results are almost the same in terms of the PSNR and the visual qualities.

4.4.5 Discussion

For discussion based on numerical values, we compare the number of iterations (Tab. 4.6) and running time (Tab. 4.7) to satisfy the stopping criteria in Tab. 4.2.

The appropriate value of the parameter for SP (γ_1) varied depending on the optimization problem and were 0.1 for mixed noise removal, 0.01 and 0.001 for unmixing, and 0.1 and 0.01 for graph signal recovery. If γ_1 is adjusted appropriately, as in the case of the unmixing experiments ($\gamma_1 = 0.01$ and 0.001), P-PDS with SP can converge faster than the automatic preconditioner design methods (ASP and OVDP). However, no parameter results in fast convergence for any optimization problem, and the convergence might be extremely slow, such as at 0.01 and 0.001 for mixed noise removal, at 1 for unmixing, and at 0.001 for graph signal recovery. Therefore, γ_1 needs to be manually adjusted according to each problem.

P-PDS with ASP was the best in terms of the average number of iterations, and P-PDS with PDP (τ is adjusted) resulted in a small number of iterations to converge for both graph signal recovery and mixed noise removal. However, for the unmixing experiments, P-PDS with PDP required a more significant number of iterations to converge than P-PDS with SP ($\gamma_1 = 0.01$ and 0.001) and OVDP. We speculate that this is because the optimization problem of unmixing is relatively complicated; it involves an endmember

matrix, while the optimization problems of mixed noise removal and graph signal recovery only include relatively simple difference operators and random sampling matrices in their optimization problems. Although P-PDSs with ASP and PDP were fast in the number of iterations, they took a much longer running time to converge. This is due to the fact that they require iterative algorithms such as FISTA to compute the skewed proximity operator in each iteration of P-PDS. Incidentally, since the internal iterations of FISTA vary depending on the task and parameters (e.g., τ), the execution time of P-PDS may be long relative to the number of iterations to convergence. For example, P-PDS with PDP ($\tau = 1$) required fewer iterations but a longer running time than P-PDS with PDP ($\tau = 0.1$). In addition, P-PDS with ASP took a very long running time per iteration in the graph signal recovery experiment, while it took a short running time in the unmixing experiments.

P-PDSs with OVDP achieved good convergence speed in both the number of iterations and the running time thanks to a diagonal preconditioning technique based on the problem structure. In addition, they maintain the proximability of the functions, resulting in fast running time. P-PDS with OVDP2 was fast on average in the number of iterations. Moreover, P-PDS with OVDP2 produced the fastest result in terms of running time for the mixed noise removal experiment. P-PDS with OVDP1 was faster than P-PDS with OVDP2 and OVDP3 for the unmixing and graph signal recovery experiments. Furthermore, the preconditioners of OVDP can be easily calculated in the mixed noise removal case whose optimization problem incorporates the linear operators implemented not as explicit matrices.

These results indicate the following conclusions.

- SP and PDP are effective for cases where preconditioners are easily adjusted. In particular, PDP is very effective for cases where the structure of an optimization problem is simple and the calculation of an inner iteration is efficient.
- ASP is applicable to the cases where the structure of an optimization problem is simple, the calculation of an inner iteration is efficient, and the optimization problem only contains linear operators implemented as the represented matrix.
- Our OVDP can determine effective preconditioners regardless of whether or not the above conditions are satisfied. Specifically, for the signal estimation problem that can be handled by ASP, our OVDP was several hundred times faster than ASP.
- In addition, P-PDS with our OVDP required fewer iterations on average than P-PDSs with SP or PDP, which require manual adjustments.

4.5 Concluding Remarks

We have proposed OVDP, which automatically and easily designs preconditioners in a variable-wise manner when a given optimization problem incorporates linear operators represented not as explicit matrices. We also proved the convergence of P-PDS with OVDP. Applications of our method to three signal estimation tasks have been provided with experimental comparison, where we have shown that our method achieved the fast convergence speed on average and raised the examples of signal processing tasks that OVDP is effective to be applied.

Table 4.6: The Number of Iterations to Meet the Stopping Criteria. XXX* Means That the Method Requires More Than XXX Iterations.

	Methods											
	SP				ASP	PDP				**OVDP1**	**OVDP2**	**OVDP3**
	($\gamma_1 = 1$)	($\gamma_1 = 0.1$)	($\gamma_1 = 0.01$)	($\gamma_1 = 0.001$)		($\tau = 1$)	($\tau = 0.1$)	($\tau = 0.01$)	($\tau = 0.001$)			
Mixed noise removal	10000*	4736	10000*	10000*	3323	10000*	1552	10000*	10000*	5470	3325	9755
Unmixing	2943	395	350	349	300	10000*	10000*	10000*	10000*	350	525	4709
Graph signal recovery	3625	806	1937	10000*	448	194	396	4129	10000*	998	1846	3546
Average	5523*	1979	4096*	6783*	1357	6731*	3983*	8043*	10000*	2273	1899	6003

Table 4.7: Running time [s] to Meet the Stopping Criteria. XXX* Means That the Method Requires More Than XXX [s].

	Methods											
	SP				ASP	PDP				**OVDP1**	**OVDP2**	**OVDP3**
	($\gamma_1 = 1$)	($\gamma_1 = 0.1$)	($\gamma_1 = 0.01$)	($\gamma_1 = 0.001$)		($\tau = 1$)	($\tau = 0.1$)	($\tau = 0.01$)	($\tau = 0.001$)			
Mixed noise removal	1000*	111.86	333.72	1000*	230.40	1000*	1000*	1000*	1000*	130.70	79.35	231.78
Unmixing	3.52	3.97	3.60	3.60	64.44	1000*	1000*	1000*	1000*	11.38	5.44	46.13
Graph signal recovery	7.54	1.65	4.00	40.61	1000*	607.90	450.60	1000*	1000*	2.04	3.73	7.21

Chapter 5

Robust Hyperspectral Unmixing Using Mixed Noise Constraint Modeling and Problem Structure-Based Stepsize Design

5.1 Introduction

Hyperspectral (HS) images are three-dimensional cube data consisting of two-dimensional spatial and one-dimensional spectral information. Compared to grayscale or RGB images, HS images provide more than several hundred bands, each of which contains specific unique wavelength characteristics of materials such as minerals, soils, and liquids. Therefore, HS images have various applications, such as ecology, mineralogy, biotechnology, and agriculture [34, 68, 120, 179]. Due to the trade-off between spatial resolution and spectral resolution, HS sensors do not have a sufficient spatial resolution, resulting in containing multiple components (called endmembers) in a pixel [95], which is referred to as a mixel. The process of decomposing the mixel into endmembers and their abundance maps is called unmixing. Unmixing has been actively studied in the remote sensing field because it is essential for HS image analysis [20,123] and other applications, such as denoising [152,216] and data fusion [111, 203].

Unmixing methods fall into two categories according to their assumptions: non-blind and blind unmixing. Non-blind unmixing methods estimate abundance maps from a given endmember library. Endmembers in the library are potentially much larger in number than endmembers included in real HS images, i.e., its corresponding abundance maps become sparse. On the other hand, blind unmixing methods simultaneously estimate an endmember library and abundance maps, allowing us to obtain the abundances of endmembers whose spectral libraries are unknown.

For blind unmixing, nonnegative matrix factorization-based approaches [63, 64, 106] and learning-based approaches [77,144,150] have attracted attention. Nonnegative matrix factorization-based methods design and solve an optimization problem that incorporates the functions of the product of an endmember matrix and an abundance map matrix. When solving the optimization problem, they take an approach that iterates alternate updates of the two matrices: updating the endmember library matrix by solving the subproblem with the abundance map matrix fixed, updating the abundance maps by solving the subproblem with the endmember library matrix fixed using some non-blind unmixing method. Learning-based methods often involve the following steps: extraction of initial endmembers from an input HS image, estimation of corresponding initial abundance

maps by some non-blind unmixing methods, and then learning of sophisticated unmixing and reconstruction networks based on this information. Therefore, non-blind unmixing is a fundamental task that must precede blind unmixing. Henceforth, non-blind unmixing will simply be referred to as unmixing.

Although very accurate unmixing can be achieved using state-of-the-art methods if a noise-free HS image is available, real-world HS images are often contaminated by various types of noise such as Gaussian noise, outliers, missing values, and stripe noise due to environmental factors and sensor failures. Such noise obviously has a negative impact on unmixing performance and needs to be dealt with appropriately. The simplest way is a two-step approach, where noise is first removed from a given HS image beforehand, followed by unmixing. However, such methods are also likely to remove even important spectral information. It is therefore essential to develop a method that can *simultaneously* separate noise (without affecting spectral information) during the unmixing process, which we refer to as *noise-robust* unmixing.

Many noise-robust unmixing techniques explicitly model noises and then take the approach of solving optimization problems that incorporate functions characterizing abundance maps. Based on the fact that HS images consist of a small fraction of the endmembers in a library, the methods in [86, 87, 105, 110, 155, 163] employ a sparse regularization. Abundance maps are also piecewise smooth because neighboring pixels often have the same endmembers. To capture the nature, the methods in [9, 88, 183, 186, 212] adopt a combination of sparse and piecewise-smooth regularizations. In addition, the methods in [149, 151, 215] estimate abundance maps using a regularization based on deep neural networks, and the methods in [81, 82, 194] adopt a combination of sparse and low-rank regularizations.

As we have discussed, various studies have been carried out to mitigate the effects of noise in unmixing, but there are still two limitations in terms of robustness to noise. The first is that the performance of unmixing is severely degraded when the input HS image is contaminated with high levels of noise. The second is that existing unmixing methods cannot adequately deal with stripe noise.

As reviewed in the previous subsection, many existing unmixing methods use a combination of sparse and piecewise-smooth regularization to characterize the abundance maps. However, as shown in Fig. 5.1, balancing these regularizations becomes very difficult when unmixing HS images contaminated with high levels of noise. In fact, if the weight of the sparse regularization is increased, a large amount of noise remains in the estimated abundance maps. Conversely, if the weight of the piecewise-smooth regularization is increased, the estimated abundance maps will contain many inappropriate components that are not present in the original HS image. In existing methods, adjusting the weights to avoid both problems is a very sensitive and tedious task.

To resolve this difficulty, we focus on the regularizations for the HS image reconstructed by mixing the estimated abundance maps and the endmembers, which we call image-domain regularizations, in addition to the regularizations for the abundance maps. Our assumption is the following: if the reconstructed HS image has desirable properties in its spatio-spectral structure, then the estimated abundance maps used for reconstruction should also have desirable properties. Therefore, we believe that incorporating spatio-spectral regularization for HS images into the unmixing formulation can improve the unmixing performance in high-noise situations where abundance maps are difficult to estimate using existing methods. Fortunately, in the context of HS image restoration, many effective HS image regularizations have been studied [8, 76, 174, 177, 184, 204, 208]. By adopting them as image-domain regularizations, we can robustify the unmixing process under highly-noisy scenarios.

Regarding the second limitation, existing unmixing methods mainly deal with Gaus-

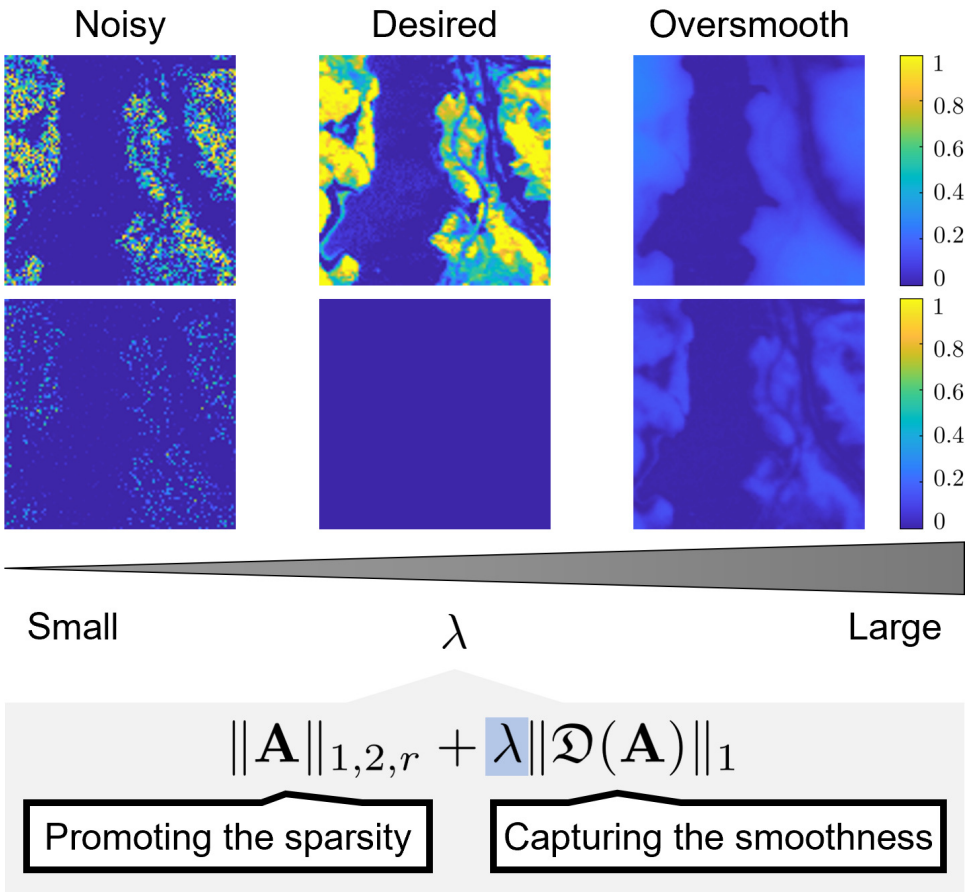


Figure 5.1: Difficulties in dealing with high intensity noise in unmixing.

sian noise and sparse noise. However, in addition to these noises, actual HS images are often contaminated with stripe noise, mainly due to external disturbances and calibration errors [28, 117, 132]. Since stripe noise is non-Gaussian and non-sparse, it cannot be handled by existing methods, leading to performance degradation in unmixing.

Based on the above discussion, we propose robust hyperspectral unmixing using image-domain regularization (RHUIDR). We formulate the unmixing problem as a constrained convex optimization problem. In order to solve the optimization problem, we develop an efficient algorithm based on the preconditioned primal-dual splitting method (P-PDS) [148] with an operator-norm-based stepsize selection method [135]. In terms of the features of RHUIDR, the contributions of the chapter can be summarized as follows.

1. *(Robust to high levels of noise):* RHUIDR employs not only the abundance map regularizations but also image-domain regularizations, which robustify the unmixing process under highly-noisy scenarios.

2. *(Robust to mixed noise including stripe noise):* By explicitly modeling three types of noise (Gaussian noise, sparse noise, and stripe noise) as in (5.9), RHUIDR can adequately handle mixed noise, including stripe noise, which is difficult to handle in existing methods.

3. *(Easy to adjust hyperparameters):* In the formulated optimization problem, we model data-fidelity and noise terms as hard constraints instead of adding them to the objective function. This type of constrained formulation decouples interdependent hyperparameters into independent ones, thus facilitating parameter settings, which will be detailed in Sec. 5.3.1.

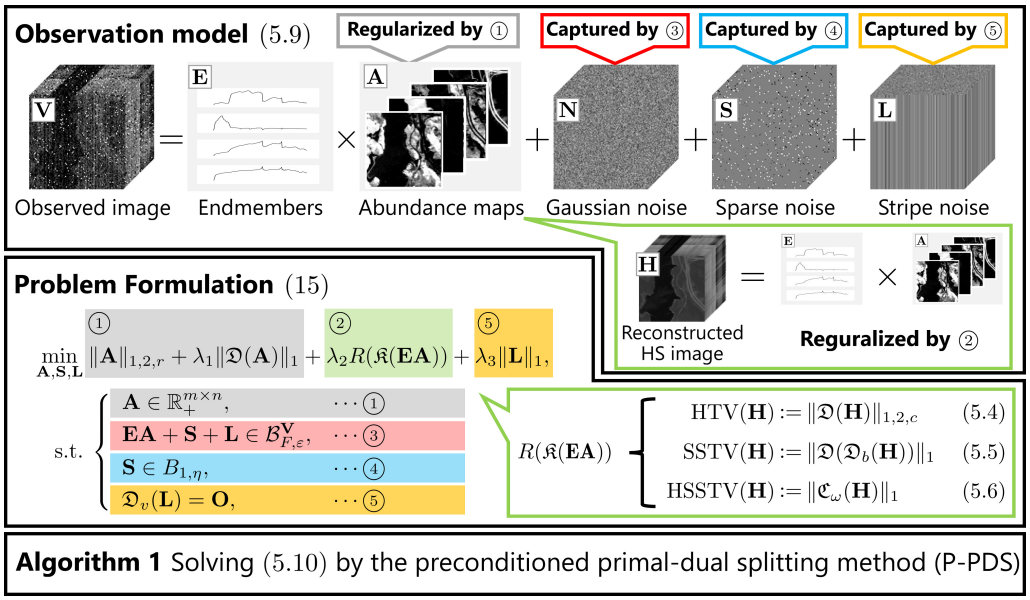


Figure 5.2: Illustration of the proposed method, i.e., RHUIDR.

4. *(Avoiding adjusting stepsizes):* Unlike the optimization algorithms used in existing unmixing methods, our P-PDS-based algorithm can automatically determine the appropriate stepsizes based on the problem structure.

Experiments on synthetic and real HS images demonstrate the advantages of RHUIDRover existing methods.

The chapter is organized as follows. In Section 5.2, we introduce mathematical tools. In Section 5.3, we explain the proposed method, RHUIDR, with its formulation and algorithm. In Section 5.4, we conduct experiments to show the superiority of RHUIDR over the existing methods. Finally, we conclude the chapter in Section 5.5.

5.2 Preliminaries

5.2.1 Regularizations for an HS Image

This section introduces the regularizations for an HS image $\mathbf{H} \in \mathbb{R}^{l\times n_1 n_2}$. Let $\mathfrak{D}_v : \mathbb{R}^{l\times n_1 n_2} \to \mathbb{R}^{l\times n_1 n_2}$, $\mathfrak{D}_h : \mathbb{R}^{l\times n_1 n_2} \to \mathbb{R}^{l\times n_1 n_2}$, and $\mathfrak{D}_b : \mathbb{R}^{l\times n_1 n_2} \to \mathbb{R}^{l\times n_1 n_2}$ be respectively vertical, horizontal, and spectral difference operators, which are given by

$$[\mathfrak{D}_v(\mathbf{H})]_{i,j,k} := \begin{cases} [\mathbf{H}]_{i+1,j,k} - [\mathbf{H}]_{i,j,k}, & \text{if } i < n_1; \\ 0, & \text{otherwise,} \end{cases} \tag{5.1}$$

$$[\mathfrak{D}_h(\mathbf{H})]_{i,j,k} := \begin{cases} [\mathbf{H}]_{i,j+1,k} - [\mathbf{H}]_{i,j,k}, & \text{if } j < n_2; \\ 0, & \text{otherwise,} \end{cases} \tag{5.2}$$

$$[\mathfrak{D}_b(\mathbf{H})]_{i,j,k} := \begin{cases} [\mathbf{H}]_{i,j,k+1} - [\mathbf{H}]_{i,j,k}, & \text{if } k < l; \\ 0, & \text{otherwise.} \end{cases} \tag{5.3}$$

Then, HTV [204], SSTV [8], and HSSTV [177] are defined by

$$\text{HTV}(\mathbf{H}) := \|\mathfrak{D}(\mathbf{H})\|_{1,2,c}, \tag{5.4}$$

$$\text{SSTV}(\mathbf{H}) := \|\mathfrak{D}(\mathfrak{D}_b(\mathbf{H}))\|_1, \tag{5.5}$$

$$\text{HSSTV}(\mathbf{H}) := \|\mathfrak{C}_\omega(\mathbf{H})\|_1, \tag{5.6}$$

where $\mathfrak{D}$ is the spatial difference operator:

$$\mathfrak{D}(\mathbf{H}) := \begin{bmatrix} \mathfrak{D}_v(\mathbf{H}) \\ \mathfrak{D}_h(\mathbf{H}) \end{bmatrix}, \tag{5.7}$$

and $\mathfrak{C}_\omega$ is a combination of spatial and spatio-spectral difference operators with a balancing parameter $\omega > 0$:

$$\mathfrak{C}_\omega(\mathbf{H}) := \begin{bmatrix} \mathfrak{D}(\mathfrak{D}_b(\mathbf{H})) \\ \omega\mathfrak{D}(\mathbf{H}) \end{bmatrix}. \tag{5.8}$$

HTV captures spectral correlations by promoting the sparsity of spatial differences grouped by the spectral direction. SSTV captures piecewise smoothness in the spatial and spectral directions by using the composite operator of the spatial and spectral differences (spatio-spectral difference). However, it does not sufficiently evaluate direct spatial piecewise smoothness, resulting in residual noise and artifacts. HSSTV promotes both spatio-spectral and direct spatial smoothness, and thus is a more powerful regularization in general.

5.3 Proposed Method

A general diagram of the proposed method, RHUIDR, is shown in Fig. 5.2. In the following, we first introduce an observation model with three types of noise. Based on the model, we then formulate the unmixing problem as a constrained convex optimization problem. Finally, we describe a P-PDS-based algorithm to efficiently solve the optimization problem with its computational complexity.

5.3.1 Problem Formulation

Let $\mathbf{E} \in \mathbb{R}^{l\times m}$, $\bar{\mathbf{A}} \in \mathbb{R}^{m\times n}$, $\bar{\mathbf{N}} \in \mathbb{R}^{l\times n}$, $\bar{\mathbf{S}} \in \mathbb{R}^{l\times n}$, and $\bar{\mathbf{L}} \in \mathbb{R}^{l\times n}$ be a given endmember library, a true abundance matrix, Gaussian noise, sparse noise, and stripe noise, respectively. Consider the following observation model:

$$\mathbf{V} = \mathbf{E}\bar{\mathbf{A}} + \bar{\mathbf{N}} + \bar{\mathbf{S}} + \bar{\mathbf{L}}. \tag{5.9}$$

Note that this model explicitly deals with stripe noise as an additive component $\bar{\mathbf{L}}$. Based on Eq. (5.9), we formulate an unmixing problem as the following constrained convex optimization problem:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{S},\mathbf{L}} \|\mathbf{A}\|_{1,2,r} + \lambda_1\|\mathfrak{D}(\mathbf{A})\|_1 + \lambda_2 R(\mathfrak{K}(\mathbf{EA})) + \lambda_3\|\mathbf{L}\|_1, \\ &\text{s.t.} \begin{cases} \mathbf{A} \in \mathbb{R}_+^{m\times n}, \\ \mathbf{EA} + \mathbf{S} + \mathbf{L} \in B_{F,\varepsilon}^{\mathbf{V}}, \\ \mathbf{S} \in B_{1,\eta}, \\ \mathfrak{D}_v(\mathbf{L}) \in \{\mathbf{O}\}, \end{cases} \end{aligned} \tag{5.10}$$

where $\lambda_1 > 0$, $\lambda_2 > 0$, and $\lambda_3 > 0$ are hyperparameters that balance each term. The first term is the joint-sparse regularization that evaluates the row sparsity of abundance maps $\mathbf{A}$. The second term promotes the piecewise smoothness of $\mathbf{A}$. The first constraint guarantees the non-negativity of $\mathbf{A}$. Note that we do not explicitly adopt the abundance sum-to-one constraint. This is because, in real-world situations, the abundance sum-to-one constraint tends to be a strong assumption for LMM-based unmixing because the spectral signatures are often affected by a positive scaling factor that varies from pixel to pixel [86].

Table 5.1: Specific Function $\mathcal{R}$ and Linear Operator $\mathfrak{K}$ in Each Rconstructed-Image Regularization

Regularizations	$\mathcal{R}$	$\mathfrak{K}$
HTV (Eq. (5.4))	$\|\cdot\|_{1,2,c}$	$\mathfrak{D}$
SSTV (Eq. (5.5))	$\|\cdot\|_1$	$\mathfrak{D}\circ\mathfrak{D}_b$
HSSTV (Eq. (5.6))	$\|\cdot\|_1$	$\mathfrak{C}_\omega$

Table 5.2: Stepsizes $\gamma_1, \gamma_2, \gamma_3$, and γ_4 for Each Algorithm That Solves An Optimization Problem Incorprating Each Reconstructed-Image Regularization.

Regularizations	γ_1	γ_2	γ_3	γ_4
HTV (Eq. (5.4))	$\frac{1}{9+9\sigma_1(\mathbf{A})^2}$	1	$\frac{1}{5}$	$\frac{1}{3}$
SSTV (Eq. (5.5))	$\frac{1}{9+33\sigma_1(\mathbf{A})^2}$	1	$\frac{1}{5}$	$\frac{1}{3}$
HSSTV (Eq. (5.6))	$\frac{1}{9+(33+8\omega^2)\sigma_1(\mathbf{A})^2}$	1	$\frac{1}{5}$	$\frac{1}{3}$

The third term is the regularization of the reconstructed HS image. This image-domain regularization allows us to avoid the sensitive adjustment of λ_1. In this chapter, we focus on three image-domain regularizations: HTV in (5.4), SSTV in (5.5), and HSSTV in (5.6). In each case, R and $\mathfrak{K}$ are defined as shown in Tab. 5.1. By further generalizing the third term, RHUIDR can incorporate other regularizations proposed, e.g., in [174, 184].

The second constraint serves as data-fidelity to the observed HS image $\mathbf{V}$ with the Frobenius norm ball $B^{\mathbf{V}}_{F,\varepsilon}$ with the center $\mathbf{V}$ and radius ε. The third constraint evaluates the sparsity of $\mathbf{S}$. As described in the third contribution, using such constraints instead of data-fidelity and sparse terms makes it easy to adjust hyperparameters since the parameters can be determined based only on noise intensity. Indeed, this kind of constrained formulation has played an important role in facilitating the parameter setup of signal recovery problems [7, 47, 137, 138, 141]. The detailed setting of these parameters ε and η is shown in Sec. V-B.

The fourth term controls the intensity of stripe noise $\mathbf{L}$ and the fourth constraint captures the vertical flatness property by imposing zero to the vertical gradient of $\mathbf{L}$. The term and constraint accurately characterize stripe noise [132]. Therefore, our method can estimate abundance maps from HS images contaminated by mixed noise including stripe noise.

5.3.2 Optimization Algorithm

To solve Prob. (5.10) by an algorithm based on P-PDS, we need to transform Prob. (5.10) into Prob. (2.28). First, using the indicator functions, Prob. (5.10) are rewritten as follows:

$$\begin{aligned}\min_{\mathbf{A},\mathbf{S},\mathbf{L}}\ & \|\mathbf{A}\|_{1,2,r} + \lambda_1\|\mathfrak{D}(\mathbf{A})\|_1 + \lambda_2 R(\mathfrak{K}\circ\mathbf{E}(\mathbf{A})) + \lambda_3\|\mathbf{L}\|_1 \\ & + \iota_{\mathbb{R}_+^{m\times n}}(\mathbf{A}) + \iota_{B^{\mathbf{V}}_{F,\varepsilon}}(\mathbf{E}\mathbf{A}+\mathbf{S}+\mathbf{L}) \\ & + \iota_{B_{1,\eta}}(\mathbf{S}) + \iota_{\{\mathbf{O}\}}(\mathfrak{D}_v(\mathbf{L})),\end{aligned} \tag{5.11}$$

where $\mathfrak{K}\circ\mathbf{E}$ is the composite operator of $\mathfrak{K}$ and $\mathbf{E}$, i.e., $\mathfrak{K}\circ\mathbf{E}(\mathbf{A}) = \mathfrak{K}(\mathbf{E}\mathbf{A})$. Introducing auxiliary variables $\mathbf{Z}_1$, $\mathbf{Z}_2$, $\mathbf{Z}_3$, $\mathbf{Z}_4$, and $\mathbf{Z}_5$, we can transform Prob. (5.11) into the following

Algorithm 3 A P-PDS-based algorithm for solving Prob. (5.10)

Input: $\mathbf{V}$, $\mathbf{E}$, λ_1, λ_2, λ_3, ε, and η
Output: $\mathbf{A}^{(t)}$, $\mathbf{S}^{(t)}$, $\mathbf{L}^{(t)}$

1: Initialize $\mathbf{A}^{(0)}$, $\mathbf{S}^{(0)}$, $\mathbf{L}^{(0)}$, $\mathbf{Z}_1^{(0)}$, $\mathbf{Z}_2^{(0)}$, $\mathbf{Z}_3^{(0)}$, $\mathbf{Z}_4^{(0)}$, and $\mathbf{Z}_5^{(0)}$;
2: Set $\gamma_1, \gamma_2, \gamma_3$, and γ_4 as in Tab. 5.2;
3: **while** until a stopping criterion is satisfied **do**
4: $\quad\mathbf{A}' \leftarrow \mathbf{A}^{(t)} - \gamma_1(\mathbf{Z}_1^{(t)} + \mathfrak{D}^*(\mathbf{Z}_2^{(t)}) + \mathbf{E}^\top(\mathfrak{K}^*(\mathbf{Z}_3^{(t)})) + \mathbf{E}^\top\mathbf{Z}_4^{(t)})$;
5: $\quad\mathbf{A}^{(t+1)} \leftarrow P_{\iota_{\mathbb{R}_+^{m\times n}}}(\mathbf{A}')$ by (2.17);
6: $\quad\mathbf{S}' \leftarrow \mathbf{S}^{(t)} - \gamma_2\mathbf{Z}_4^{(t)}$;
7: $\quad\mathbf{S}^{(t+1)} \leftarrow \mathrm{prox}_{\gamma_2\iota_{B_{1,\eta}}}(\mathbf{S}')$ by the method in [53];
8: $\quad\mathbf{L}' \leftarrow \mathbf{L}^{(t)} - \gamma_3(\mathbf{Z}_4^{(t)} + \mathfrak{D}_v^*(\mathbf{Z}_5^{(t)}))$;
9: $\quad\mathbf{L}^{(t+1)} \leftarrow \mathrm{prox}_{\gamma_3\lambda_3\|\cdot\|_1}(\mathbf{L}')$ by (2.23);
10: $\quad\mathbf{Z}_1' \leftarrow \mathbf{Z}_1^{(t)} + \gamma_4(2\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)})$;
11: $\quad\mathbf{Z}_1^{(t+1)} \leftarrow \mathbf{Z}_1' - \gamma_4\mathrm{prox}_{\frac{1}{\gamma_4}\|\cdot\|_{1,2,r}}(\frac{\mathbf{Z}_1'}{\gamma_4})$ by (2.25);
12: $\quad\mathbf{Z}_2' \leftarrow \mathbf{Z}_2^{(t)} + \gamma_4\mathfrak{D}(2\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)})$;
13: $\quad\mathbf{Z}_2^{(t+1)} \leftarrow \mathbf{Z}_2' - \gamma_4\mathrm{prox}_{\frac{\lambda_1}{\gamma_4}\|\cdot\|_1}(\frac{\mathbf{Z}_2'}{\gamma_4})$ by (2.23);
14: $\quad\mathbf{Z}_3' \leftarrow \mathbf{Z}_3^{(t)} + \gamma_4\mathfrak{K}(\mathbf{E}(2\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}))$;
15: $\quad\mathbf{Z}_3^{(t+1)} \leftarrow \mathbf{Z}_3' - \gamma_4\mathrm{prox}_{\frac{\lambda_2}{\gamma_4}R}(\frac{\mathbf{Z}_3'}{\gamma_4})$ by (2.23) or (2.26);
16: $\quad\mathbf{Z}_4' \leftarrow \mathbf{Z}_4^{(t)} + \gamma_4(2(\mathbf{E}\mathbf{A}^{(t+1)} + \mathbf{S}^{(t+1)} + \mathbf{L}^{(t+1)}) - (\mathbf{E}\mathbf{A}^{(t)} + \mathbf{S}^{(t)} + \mathbf{L}^{(t)}))$;
17: $\quad\mathbf{Z}_4^{(t+1)} \leftarrow \mathbf{Z}_4' - \gamma_4\mathrm{prox}_{\frac{1}{\gamma_4}\iota_{B_{F,\varepsilon}^{\mathbf{V}}}}(\frac{\mathbf{Z}_4'}{\gamma_4})$ by (2.16);
18: $\quad\mathbf{Z}_5' \leftarrow \mathbf{Z}_5^{(t)} + \gamma_4\mathfrak{D}_v(2\mathbf{L}^{(t+1)} - \mathbf{L}^{(t)})$;
19: $\quad\mathbf{Z}_5^{(t+1)} \leftarrow \mathbf{Z}_5' - \gamma_4\mathrm{prox}_{\frac{1}{\gamma_4}\iota_{\{\mathbf{O}\}}}(\frac{\mathbf{Z}_5'}{\gamma_4})$ by (2.18);
20: $\quad t \leftarrow t + 1$;
21: **end while**

equivalent problem:

$$\begin{aligned}\min_{\substack{\mathbf{A},\mathbf{S},\mathbf{L},\\ \mathbf{Z}_1,\ldots,\mathbf{Z}_5}}\ & \iota_{\mathbb{R}_+^{m\times n}}(\mathbf{A}) + \iota_{B_{1,\eta}}(\mathbf{S}) + \lambda_3\|\mathbf{L}\|_1 + \|\mathbf{Z}_1\|_{1,2,r}\\ &+ \lambda_1\|\mathbf{Z}_2\|_1 + \lambda_2 R(\mathbf{Z}_3) + \iota_{B_{F,\varepsilon}^{\mathbf{V}}}(\mathbf{Z}_4) + \iota_{\{\mathbf{O}\}}(\mathbf{Z}_5)\\ \text{s.t.}\ & \begin{cases}\mathbf{Z}_1 = \mathbf{A},\\ \mathbf{Z}_2 = \mathfrak{D}(\mathbf{A}),\\ \mathbf{Z}_3 = \mathfrak{K}\circ\mathbf{E}(\mathbf{A}),\\ \mathbf{Z}_4 = \mathbf{E}\mathbf{A} + \mathbf{S} + \mathbf{L},\\ \mathbf{Z}_5 = \mathfrak{D}_v(\mathbf{L}).\end{cases}\end{aligned} \tag{5.12}$$

Finally, by defining $f_1(\mathbf{A}) = \iota_{\mathbb{R}_+^{m\times n}}(\mathbf{A})$, $f_2(\mathbf{S}) = \iota_{B_{1,\eta}}(\mathbf{S})$, $f_3(\mathbf{L}) = \lambda_3\|\mathbf{L}\|_1$, $g_1(\mathbf{Z}_1) = \|\mathbf{Z}_1\|_{1,2,r}$, $g_2(\mathbf{Z}_2) = \lambda_1\|\mathbf{Z}_2\|_1$, $g_3(\mathbf{Z}_3) = \lambda_2 R(\mathbf{Z}_3)$, $g_4(\mathbf{Z}_4) = \iota_{B_{F,\varepsilon}^{\mathbf{V}}}(\mathbf{Z}_4)$, and $g_5(\mathbf{Z}_5) = \iota_{\{\mathbf{O}\}}(\mathbf{Z}_5)$, Prob. (2.28) is reduced to Prob. (5.12), i.e., Prob. (5.10).

The algorithm for solving Prob. (5.10) is summarized in Algorithm 3. The linear operator $\mathfrak{K}$ in steps 4 and 14, and the function R in step 15 depend on what regularization is adopted, as shown in Tab. 5.1.

Table 5.3: Computational Complexities of Each Operation.

Operations	O-notation
$\mathbf{EA}$, ($\mathbf{E} \in \mathbb{R}^{l\times m}$ and $\mathbf{A} \in \mathbb{R}^{m\times n}$)	$O(nml)$
$\mathfrak{D}(\mathbf{A})$, ($\mathbf{A} \in \mathbb{R}^{m\times n}$)	$O(nm)$
$\mathfrak{D}(\mathbf{H})$, ($\mathbf{H} \in \mathbb{R}^{l\times n}$)	$O(nl)$
$\mathfrak{K}(\mathbf{H})$, $\begin{cases}\mathfrak{K} = \mathfrak{D}, (\mathbf{H} \in \mathbb{R}^{l\times n}) \\ \mathfrak{K} = \mathfrak{D} \circ \mathfrak{D}_b, (\mathbf{H} \in \mathbb{R}^{l\times n}) \\ \mathfrak{K} = \mathfrak{C}_\omega, (\mathbf{H} \in \mathbb{R}^{l\times n})\end{cases}$	$O(nl)$
$\mathrm{prox}_{\gamma_1 \iota_{\mathbb{R}_+^{m\times n}}}(\mathbf{A})$, ($\mathbf{A} \in \mathbb{R}^{m\times n}$)	$O(nm)$
$\mathrm{prox}_{\gamma_2 \iota_{B_{1,\eta}}}(\mathbf{S})$, ($\mathbf{S} \in \mathbb{R}^{l\times n}$)	$O(nl \log nl)$
$\mathrm{prox}_{\gamma_3\lambda_3\|\cdot\|_1}(\mathbf{L})$, ($\mathbf{L} \in \mathbb{R}^{l\times n}$)	$O(nl)$
$\mathrm{prox}_{\frac{1}{\gamma_4}\|\cdot\|_{1,2,r}}(\mathbf{Z}_1)$, ($\mathbf{Z}_1 \in \mathbb{R}^{m\times n}$)	$O(nm)$
$\mathrm{prox}_{\frac{\lambda_1}{\gamma_4}\|\cdot\|_1}(\mathbf{Z}_2)$, ($\mathbf{Z}_2 \in \mathbb{R}^{2m\times n}$)	$O(nm)$
$\mathrm{prox}_{\frac{\lambda_2}{\gamma_4}R}(\mathbf{Z}_3)$, $\begin{cases}R = \|\cdot\|_{1,2,c}, (\mathbf{Z}_3 \in \mathbb{R}^{2l\times n}) \\ R = \|\cdot\|_1, (\mathbf{Z}_3 \in \mathbb{R}^{2l\times n}) \\ R = \|\cdot\|_1, (\mathbf{Z}_3 \in \mathbb{R}^{4l\times n})\end{cases}$	$O(nl)$
$\mathrm{prox}_{\frac{1}{\gamma_4}\iota_{B^{\mathbf{V}}_{F,\varepsilon}}}(\mathbf{Z}_4)$, ($\mathbf{Z}_4 \in \mathbb{R}^{l\times n}$)	$O(nl)$

Based on Eq. (4.12), the stepsizes of Algorithm 3 are given as

$$\begin{aligned}&\gamma_1 = \tfrac{1}{\|\mathbf{I}\|_{\mathrm{op}}^2+\|\mathfrak{D}\|_{\mathrm{op}}^2+\|\mathfrak{K}\circ\mathbf{E}\|_{\mathrm{op}}^2+\|\mathbf{E}\|_{\mathrm{op}}^2}, \gamma_2 = \tfrac{1}{\|\mathbf{I}\|_{\mathrm{op}}^2}, \\ &\gamma_3 = \tfrac{1}{\|\mathbf{I}\|_{\mathrm{op}}^2+\|\mathfrak{D}_v\|_{\mathrm{op}}^2}, \gamma_4 = \tfrac{1}{3}.\end{aligned} \tag{5.13}$$

An identity matrix of any size satisfies $\|\mathbf{I}\|_{\mathrm{op}} = 1$. The operator norm $\|\mathbf{E}\|_{\mathrm{op}}$ is equal to its maximum singular value $\sigma_1(\mathbf{E})$. The operator norms of the other linear operators are not easily obtained[1], but they are suppressed by $\|\mathfrak{D}_v\|_{\mathrm{op}} \leq 2$, $\|\mathfrak{D}_b\|_{\mathrm{op}} \leq 2$, $\|\mathfrak{D}\|_{\mathrm{op}} \leq 2\sqrt{2}$, $\|\mathfrak{K} \circ \mathbf{E}\|_{\mathrm{op}} \leq \|\mathfrak{K}\|_{\mathrm{op}}\|\mathbf{E}\|_{\mathrm{op}}$, $\|\mathfrak{D} \circ \mathfrak{D}_b\|_{\mathrm{op}} \leq \|\mathfrak{D}\|_{\mathrm{op}}\|\mathfrak{D}_b\|_{\mathrm{op}}$, and $\|\mathfrak{C}_\omega\|_{\mathrm{op}} \leq \sqrt{32+8\omega^2}$. By substituting these upper bounds into Eq. (4.12), the specific stepsizes are given as shown in Tab. 5.2. This stepsizes design method allows us to avoid the stepsize adjustment for Algorithm 3.

5.3.3 Computational Complexity

In general, the computational complexity of our algorithm varies depending on what function and linear operator are used as an image-domain regularization. Our method adopts three image-domain regularizations: HTV, SSTV, and HSSTV. The computational complexities of linear operators and functions including all the image-domain regularizations are given in Tab. 5.3. From these results, we derive the computational complexities of each step as follows:

- The complexities of Steps 4, 14, and 16 are $O(nml)$.

[1]Note that the difference operators are not implemented as matrices. Therefore, we cannot easily obtain the singular values of the matrices representing the difference operators.

Table 5.4: Noise Taken Into Account by Each Method.

Methods	Gaussian noise	Sparse noise	Stripe noise
CLSUnSAL [87]	✓	-	-
JSTV [9]	✓	✓	-
RSSUn-TV [183]	✓	-	-
LGSU [163]	✓	-	-
UnDIP [151]	✓	-	-
MdLRR [194]	✓	-	-
RHUIDR (Ours)	✓	✓	✓

- The complexities of Steps 5, 10, 11, 12, and 13 are $O(nm)$.
- The complexities of Steps 6, 8, 9, 15, 17, 18, and 19 are $O(nl)$.
- The complexity of Step 7 is $O(nl \log nl)$.

Therefore, the complexity for each iteration of the algorithm is $O(nl \max\{m, \log nl\})$.

5.4 Experiments

We demonstrate the effectiveness of the proposed non-blind unmixing method, i.e., RHUIDR through comprehensive experiments using two synthetic and two real HS images. Specifically, these experiments aim to validate that

- RHUIDR achieves good unmixing performance due to image-domain regularizations,
- RHUIDR is robust to mixed noise, including stripe noise.

As described in the introduction, existing unmixing methods are classified into blind and non-blind, depending on whether the endmember library is given or not. Due to the different assumptions and the fact that blind unmixing methods require a non-blind unmixing step to obtain an initial estimate, it is difficult to fairly compare non-blind unmixing methods with blind ones. Therefore, we compare RHUIDR with five state-of-the-art non-blind unmixing methods: the collaborative sparse unmixing by variable splitting and augmented lagrangian (CLSUnSAL) [87], the hyperspectral unmixing using joint-sparsity and total variation (JSTV) [9], the row-sparsity spectral unmixing via total variation (RSSUn-TV) [183], the local-global-based sparse regression unmixing (LGSU) [163], the hyperspectral unmixing using deep image prior (UnDIP) [151], and the multidimensional low-rank representation-based sparse hyperspectral unmixing (MdLRR) [194]. Tab. 5.4 shows the types of noise taken into account by these methods.

5.4.1 Data Description

We used four datasets for experiments. In all datasets, their endmember libraries were composed of the spectral signatures of the endmembers in ground-truth HS images and other spectral signatures. This is to simulate the real-world situation where we give an endmember library by including more spectral signatures than the components of the target HS image, as assumed in many references of non-blind unmixing.

Table 5.5: SRE, RMSE, Ps, MPSNR, and MSSIM in the Experiments Using *Synth 1*.

Noise	Evaluation Target	Measures	Methods								
			CLSUnSAL [87]	JSTV [9]	RSSUn-TV [183]	LGSU [163]	UnDIP [151]	MdLRR [194]	**RHUIDR (HTV)**	**RHUIDR (SSTV)**	**RHUIDR (HSSTV)**
Case 1	Abundance	SRE	16.50	12.95	11.39	16.59	0.58	17.98	**27.22**	27.02	26.99
		RMSE	0.0590	0.0840	0.0992	0.0608	0.3414	0.0524	**0.0173**	0.0177	0.0177
		Ps	0.68	0.56	0.53	0.64	0.18	0.66	**0.92**	**0.92**	**0.92**
	Reconstructed HS image	MPSNR	41.26	39.94	39.70	41.19	11.80	41.90	46.09	44.49	**46.26**
		MSSIM	0.9771	0.9706	0.9693	0.9754	0.4659	0.9821	0.9936	0.9906	**0.9940**
Case 2	Abundance	SRE	12.81	7.83	7.13	12.73	0.88	16.71	21.85	21.84	**22.12**
		RMSE	0.0889	0.1424	0.1530	0.0923	0.3302	0.0588	0.0318	0.0318	**0.0308**
		Ps	0.55	0.41	0.40	0.52	0.19	0.63	0.80	0.80	**0.81**
	Reconstructed HS image	MPSNR	36.13	34.51	34.42	35.39	12.68	36.07	38.86	37.91	**39.19**
		MSSIM	0.9322	0.9114	0.9101	0.9213	0.4138	0.9327	0.9660	0.9585	**0.9692**
Case 3	Abundance	SRE	11.42	11.97	5.72	10.72	0.87	13.84	**26.40**	26.04	26.07
		RMSE	0.1018	0.0931	0.1720	0.1138	0.3305	0.0802	**0.0190**	0.0197	0.0196
		Ps	0.53	0.53	0.36	0.49	0.18	0.58	**0.91**	0.90	**0.91**
	Reconstructed HS image	MPSNR	33.82	39.02	31.94	32.73	13.36	33.25	45.43	43.64	**45.58**
		MSSIM	0.9026	0.9641	0.8688	0.8821	0.3607	0.8943	0.9926	0.9885	**0.9930**
Case 4	Abundance	SRE	9.20	10.87	3.47	7.18	0.68	9.79	25.76	25.61	**25.81**
		RMSE	0.1275	0.1045	0.2088	0.1614	0.3378	0.1232	0.0205	0.0208	**0.0203**
		Ps	0.46	0.51	0.29	0.39	0.21	0.47	**0.89**	**0.89**	**0.90**
	Reconstructed HS image	MPSNR	30.73	38.20	28.88	29.28	13.11	29.75	43.69	42.38	**43.97**
		MSSIM	0.8505	0.9573	0.8003	0.8115	0.3799	0.8259	0.9884	0.9844	**0.9894**
Case 5	Abundance	SRE	10.68	7.86	4.85	9.61	0.56	12.04	23.85	23.71	**24.00**
		RMSE	0.1096	0.1418	0.1860	0.1264	0.3424	0.0965	0.0252	0.0255	**0.0247**
		Ps	0.50	0.43	0.34	0.46	0.16	0.53	0.82	0.82	**0.84**
	Reconstructed HS image	MPSNR	32.93	35.52	31.16	31.81	11.64	32.37	41.33	40.13	**41.76**
		MSSIM	0.8908	0.9338	0.8534	0.8649	0.4211	0.8790	0.9850	0.9792	**0.9870**
Case 6	Abundance	SRE	9.79	5.83	4.16	8.10	0.99	11.05	**20.13**	19.72	19.63
		RMSE	0.1209	0.1720	0.1981	0.1471	0.3259	0.1075	**0.0377**	0.0393	0.0396
		Ps	0.47	0.36	0.31	0.42	0.23	0.50	0.79	**0.80**	0.82
	Reconstructed HS image	MPSNR	31.85	32.35	30.13	30.62	13.87	31.19	38.85	36.95	**39.44**
		MSSIM	0.8615	0.8702	0.8191	0.8294	0.4129	0.8442	0.9702	0.9527	**0.9753**

Synthetic HS Image 1 (*Synth 1*)

We generated the first synthetic HS image with the size of $64 \times 64 \times 224$ using the HYperspectral Data Retrieval and Analysis (HYDRA) toolbox[2], which was developed by the Computational Intelligence group at the University of the Basque Country. An endmember library consists of 10 spectral signatures with 224 bands from the U.S. Geological Survey (USGS) Spectral Library[3]. From the endmember library, we generated the corresponding four original abundance maps with the spatial size of 64×64 using the Legendre method. Figure 5.3 (a) shows one band of the generated image.

Synthetic HS Image 2 (*Synth 2*)

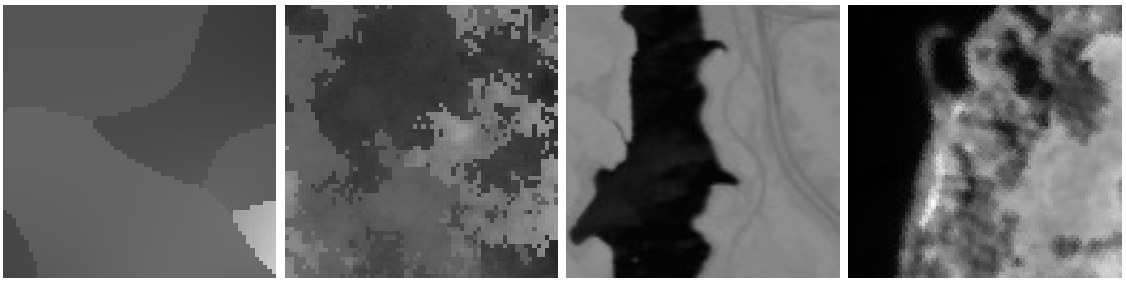
We also generated the second synthetic HS image with a size of $64 \times 64 \times 224$ using the HYDRA toolbox. An endmember library consists of 10 spectral signatures with 224 bands from the USGS Spectral Library. From the endmember library, we randomly selected four endmembers and generated the corresponding four original abundance maps with a spatial size of 64×64 using the spherical Gaussian method. Figure 5.3 (b) shows one band of the generated image.

Real Image 1 (*Jasper Ridge*)

Jasper Ridge image (see Fig. 5.3 (c)) is captured using an AVIRIS sensor in a rural area in California, USA. The spatial size of the original data is 512×614 pixels, and each pixel

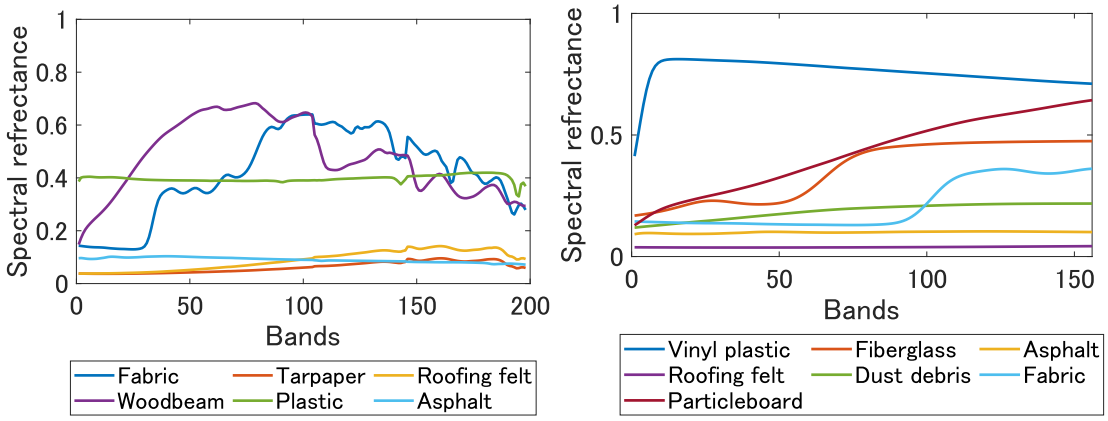
[2] `https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Imagery_Synthesis_tools_for_MATLAB`, accessed on Feb. 5, 2023

[3] `https://www.usgs.gov/programs/usgs-library`, accessed on Aug. 7, 2023



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

Figure 5.3: Original HS images. (a): A synthetic image generated using a Legendre method (*Synth 1*). (b): A synthetic image generated using a spherical Gaussian method (*Synth 2*). (c): A real image (*Jasper Ridge*). (d): A real image (*Samson*).



(a): *Jasper Ridge* (b): *Samson*

Figure 5.4: Spectral signatures added into endmember libraries of real image experiments.

holds spectral information in 224 bands ranging from 380 nm to 2500 nm. After removing several noisy bands and cropping the image, we obtained the image with 100×100 pixels and 198 bands. *Jasper Ridge* contains four major endmembers[4]: "road," "soil," "water," and "tree" [157]. Adding the six endmembers from the USGS Spectral Library, we used 10 endmembers for the experiments.

Real Image 2 (*Samson*)

Samson (see Fig. 5.3 (d)) is often used for unmixing. The spatial size of the original data is 952×952 pixels, and each pixel holds spectral information in 156 bands covering the wavelengths from 401 nm to 889 nm. After cropping the image, we obtained the image with 95×95 pixels. *Samson* contains three major endmembers: "soil," "tree," and "water." Adding the seven endmembers from the USGS Spectral Library, we used 10 endmembers for the experiments.

5.4.2 Experimental Setup

HS images are often degraded by mixed noise in real-noise scenarios. Thus, we consider the following six combinations of Gaussian noise with different standard deviations σ, salt-and-pepper noise with a different rate $p_{\mathbf{S}}$, and stripe noise in both synthetic and real data experiments.

[4]True abundance maps are available from `http://lesun.weebly.com/hyperspectral-data-set.html`

Case 1: The observed HS image is contaminated by white Gaussian noise with the standard deviation $\sigma = 0.05$.

Case 2: The observed HS image is contaminated by white Gaussian noise with the standard deviation $\sigma = 0.1$.

Case 3: The observed HS image is contaminated by white Gaussian noise with the standard deviation $\sigma = 0.05$ and salt-and-pepper noise with the rate $p_{\mathbf{S}} = 0.05$.

Case 4: The observed HS image is contaminated by white Gaussian noise with the standard deviation $\sigma = 0.05$ and salt-and-pepper noise with the rate $p_{\mathbf{S}} = 0.1$.

Case 5: The observed HS image is contaminated by white Gaussian noise with the standard deviation $\sigma = 0.05$ and salt-and-pepper noise with the rate $p_{\mathbf{S}} = 0.05$. In addition, the observed HS image is corrupted by vertical stripe noise whose intensity is random in the range $[-0.3, 0.3]$.

Case 6: The observed HS image is contaminated by white Gaussian noise with the standard deviation $\sigma = 0.1$ and salt-and-pepper noise with the rate $p_{\mathbf{S}} = 0.05$. In addition, the observed HS image is corrupted by vertical stripe noise whose intensity is random in the range $[-0.3, 0.3]$.

As the parameters of existing methods, we used the values recommended in each reference. The parameters of RHUIDR were set as $\lambda_1 = 2, \lambda_2 = 2$, and $\lambda_3 = 1$, which are adjusted to achieve the best performance. As the parameter of HSSTV, we adopted $\omega = 0.05$, which is recommended in [177]. The parameters η and ε were set as $\eta = 0.45 p_{\mathbf{S}} nl$ and $\varepsilon = 0.95\sigma\sqrt{(1 - p_{\mathbf{S}})nl}$. The maximum iteration and the stopping criterion were set to $100,000$ and $\|\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}\|_F / \|\mathbf{A}^{(t+1)}\|_F \leq 10^{-5}$, respectively.

For the quantitative evaluation of abundance maps, we used the signal reconstruction error (SRE):

$$\mathrm{SRE[dB]} = 10 \log_{10} \left(\frac{\|\bar{\mathbf{A}}\|_F^2}{\|\bar{\mathbf{A}} - \hat{\mathbf{A}}\|_F^2} \right), \tag{5.14}$$

the root-mean-square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\tfrac{1}{mn} \|\bar{\mathbf{A}} - \hat{\mathbf{A}}\|_F^2}, \tag{5.15}$$

and the probability of success (Ps):

$$\mathrm{Ps} = P\left(\tfrac{(\bar{A}_{i,j} - \hat{A}_{i,j})^2}{(\bar{A}_{i,j})^2} \leq \mathrm{threshold} \right), \tag{5.16}$$

where $\bar{\mathbf{A}}$ and $\hat{\mathbf{A}}$ denote the true and estimated abundance maps, respectively. SRE and RMSE evaluate the difference between the true and estimated abundance maps, with larger SRE or smaller RMSE indicating better-estimated performance. Ps is the probability that the relative error is less than a certain threshold. In this case, the threshold was set at 0.316, which is 5 dB.

For the quantitative evaluation of the reconstructed HS images, we used the mean peak signal-to-noise ratio overall bands (MPSNR):

$$\mathrm{MPSNR[dB]} = \frac{1}{l} \sum_{i=1}^{l} 10 \log_{10} \left(\frac{n}{\|\bar{H}_{i,j} - \hat{H}_{i,j}\|_F^2} \right), \tag{5.17}$$

where $\bar{\mathbf{H}}$ and $\hat{\mathbf{H}}$ are the ground-truth and reconstructed HS images, respectively. In addition, we adopted the mean structural similarity overall bands (MSSIM) [188]:

$$\mathrm{MSSIM} = \frac{1}{l} \sum_{i=1}^{l} \mathrm{SSIM}(\bar{\mathbf{H}}_i, \hat{\mathbf{H}}_i), \tag{5.18}$$

where $\bar{\mathbf{H}}_i$ are $\hat{\mathbf{H}}_i$ are the ith bands of $\bar{\mathbf{H}}$ and $\hat{\mathbf{H}}$, respectively. Higher MPSNR and MSSIM values indicate better reconstruction results.

5.4.3 Experimental Results With Synthetic HS Images

Tables 5.5 and 5.6 show the SRE, RMSE, Ps, MPSNR, and MSSIM results with the synthetic HS images generated using the Legendre method and spherical Gaussian methods, respectively. The best and second-best results are highlighted in bold and underlined, respectively. JSTV and RSSUn-TV were not good in all cases. CLSUnSAL was best in SRE and RMSE for Case 1 using *Synth 2*, but degraded its performance as σ increased (see Case 2 of *Synth 2*). In other cases, LGSU degraded its performance. In addition, UnDIP yielded the worst results. This is because UnDIP does not capture the sparsity of abundance maps. In contrast, RHUIDR yielded the best SRE, RMSE, Ps, MPSNR, and MSSIM values for the cases where the HS image is contaminated with noise that can be handled by the existing methods (Cases 1 and 2 for CLSUnSAL, RSSUn-TV, and LGSU, and Cases 1, 2, 3, and 4 for JSTV), except for Case 1 using *Synth 1*. This indicates that the image-domain regularizations can improve the unmixing performance. In addition, RHUIDR achieved the best performance for the other cases (Cases 5 and 6). This is due to the fact that RHUIDR can handle all three types of noise. When comparing the image-domain regularizations, SSTV and HSSTV performed better.

Figures 5.5 and 5.6 show the estimated abundance maps of *Synth 1* in Case 2 and Case 4, respectively. Figure 5.7 depicts the estimated abundance maps for *Synth 2* in Case 5. All the abundance maps of CLSUnSAL, JSTV, RSSUn-TV, and LGSU include residual noise in Cases 2, 4, and 5 (see Figs. 5.5, 5.6, and 5.7 (b)-(e)). The abundance maps of MdLRR are relatively exact in Case 2 (Fig. (5.5) (g)), but are affected by sparse (and stripe noise) in Cases 4 and 5 (Figs.5.6 and 5.7 (g)). UnDIP erroneously estimated that the abundances were high for the endmembers that are not present in the HS images due to the insufficient ability to capture the sparsity of abundance maps, resulting in the generation of inappropriate abundance maps (see Figs. 5.5, 5.6, and 5.7 (f)). In particular, all of the existing methods do not take stripe noise into account, producing abundance maps greatly affected by stripe noise (see Fig. 5.7 (b)-(g)). In contrast, RHUIDR exactly estimated abundance maps regardless of what type of noise contaminates HS images (see Figs. 5.5, 5.6, and 5.7 (h)-(j)).

Figures 5.8 and 5.9 display the reconstructed HS images of *Synth 1* in Case 2 and Case 4, respectively. Figure 5.10 depicts the reconstructed HS images of *Synth 2* in Case 5. All of the existing methods resulted in Gaussian noise remaining in the reconstructed HS images in Case 2 (see Fig. 5.8 (c)-(h)). Moreover, in the reconstructed HS images in Case 5 (see Fig. 5.10 (c)-(h)), we can see that residual stripe noise remains. On the other hand, RHUIDR produced clean reconstructed HS images due to the image-domain regularizations.

Table 5.6: SRE, RMSE, Ps, MPSNR, and MSSIM in the Experiments Using *Synth 2*.

Noise	Evaluation Target	Measures	Methods								
			CLSUnSAL [87]	JSTV [9]	RSSUn-TV [183]	LGSU [163]	UnDIP [151]	MdLRR [194]	**RHUIDR (HTV)**	**RHUIDR (SSTV)**	**RHUIDR (HSSTV)**
Case 1	Abundance	SRE	**10.57**	9.39	8.70	6.50	5.89	6.45	9.95	9.98	9.96
		RMSE	**0.1081**	0.1239	0.1342	0.1694	0.1853	0.1681	0.1162	0.1158	0.1160
		Ps	0.60	0.52	0.49	0.54	0.35	0.55	0.78	**0.79**	**0.79**
	Reconstructed HS image	MPSNR	40.41	39.58	39.38	40.45	21.80	40.80	42.12	41.69	**42.16**
		MSSIM	0.9870	0.9841	0.9833	0.9875	0.8409	0.9891	**0.9926**	0.9912	0.9925
Case 2	Abundance	SRE	7.97	6.00	5.64	5.74	5.55	5.92	8.39	8.45	**8.48**
		RMSE	0.1459	0.1829	0.1907	0.1837	0.1928	0.1789	0.1390	0.1381	**0.1376**
		Ps	0.49	0.39	0.38	0.45	0.34	0.46	0.61	0.62	**0.63**
	Reconstructed HS image	MPSNR	35.10	34.10	34.01	34.88	22.02	35.32	36.25	36.01	**36.51**
		MSSIM	0.9593	0.9485	0.9477	0.9562	0.8005	0.9612	0.9715	0.9693	**0.9732**
Case 3	Abundance	SRE	7.19	8.67	4.27	5.11	2.95	5.19	9.73	**9.74**	9.72
		RMSE	0.1596	0.1346	0.2233	0.1986	0.2601	0.1938	0.1192	**0.1190**	0.1192
		Ps	0.47	0.50	0.34	0.42	0.35	0.42	0.78	**0.79**	0.78
	Reconstructed HS image	MPSNR	32.88	38.63	31.60	32.24	21.17	32.56	41.30	40.89	**41.40**
		MSSIM	0.9423	0.9806	0.9235	0.9331	0.8355	0.9387	**0.9914**	0.9898	**0.9914**
Case 4	Abundance	SRE	6.22	8.43	3.02	3.73	2.28	5.00	9.70	**9.77**	9.76
		RMSE	0.1785	0.1384	0.2579	0.2465	0.2810	0.1997	0.1195	**0.1186**	0.1187
		Ps	0.43	0.48	0.29	0.36	0.28	0.38	0.77	**0.78**	**0.78**
	Reconstructed HS image	MPSNR	29.97	37.94	28.63	28.98	20.97	29.20	40.63	40.20	**40.75**
		MSSIM	0.9093	0.9773	0.8775	0.8876	0.7972	0.8936	0.9898	0.9880	**0.9899**
Case 5	Abundance	SRE	6.84	6.36	4.01	4.77	2.77	5.42	9.00	**9.02**	8.99
		RMSE	0.1661	0.1756	0.2301	0.2108	0.2654	0.1880	0.1295	**0.1293**	0.1297
		Ps	0.45	0.41	0.32	0.40	0.21	0.41	0.74	**0.75**	0.74
	Reconstructed HS image	MPSNR	32.34	35.46	31.00	31.55	19.63	31.80	39.10	38.40	**39.32**
		MSSIM	0.9368	0.9640	0.9150	0.9242	0.7990	0.9295	0.9880	0.9850	**0.9885**
Case 6	Abundance	SRE	6.46	4.62	3.37	4.16	4.16	5.27	7.94	**7.97**	7.92
		RMSE	0.1736	0.2146	0.2477	0.2298	0.2261	0.1917	0.1464	**0.1459**	0.1468
		Ps	0.43	0.34	0.30	0.37	0.30	0.40	0.68	**0.69**	0.68
	Reconstructed HS image	MPSNR	31.18	32.12	29.95	30.38	21.76	30.61	35.97	35.09	**36.11**
		MSSIM	0.9175	0.9246	0.8926	0.9012	0.8032	0.9066	0.9735	0.9641	**0.9737**

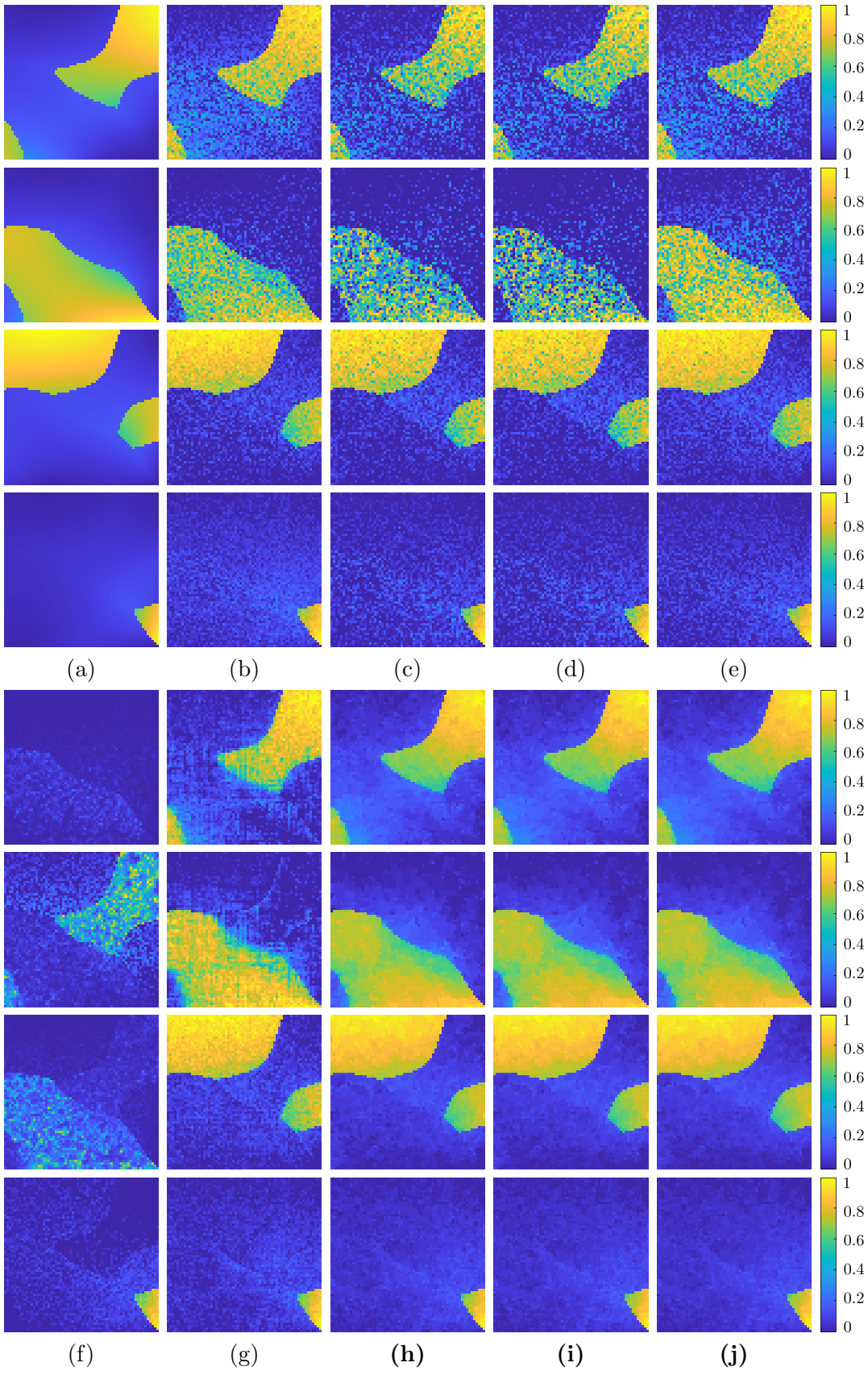


Figure 5.5: Unmixing results of abundance maps for the *Synth 1* experiments in Case 2. (a): Original abundance maps. (b): CLSUnSAL [87]. (c): JSTV [9]. (d): RSSUnTV [183]. (e): LGSU [163]. (f): UnDIP [151]. (g): MdLRR [194]. (h): **RHUIDR (HTV)**. (i): **RHUIDR (SSTV)**. (j): **RHUIDR (HSSTV)**.

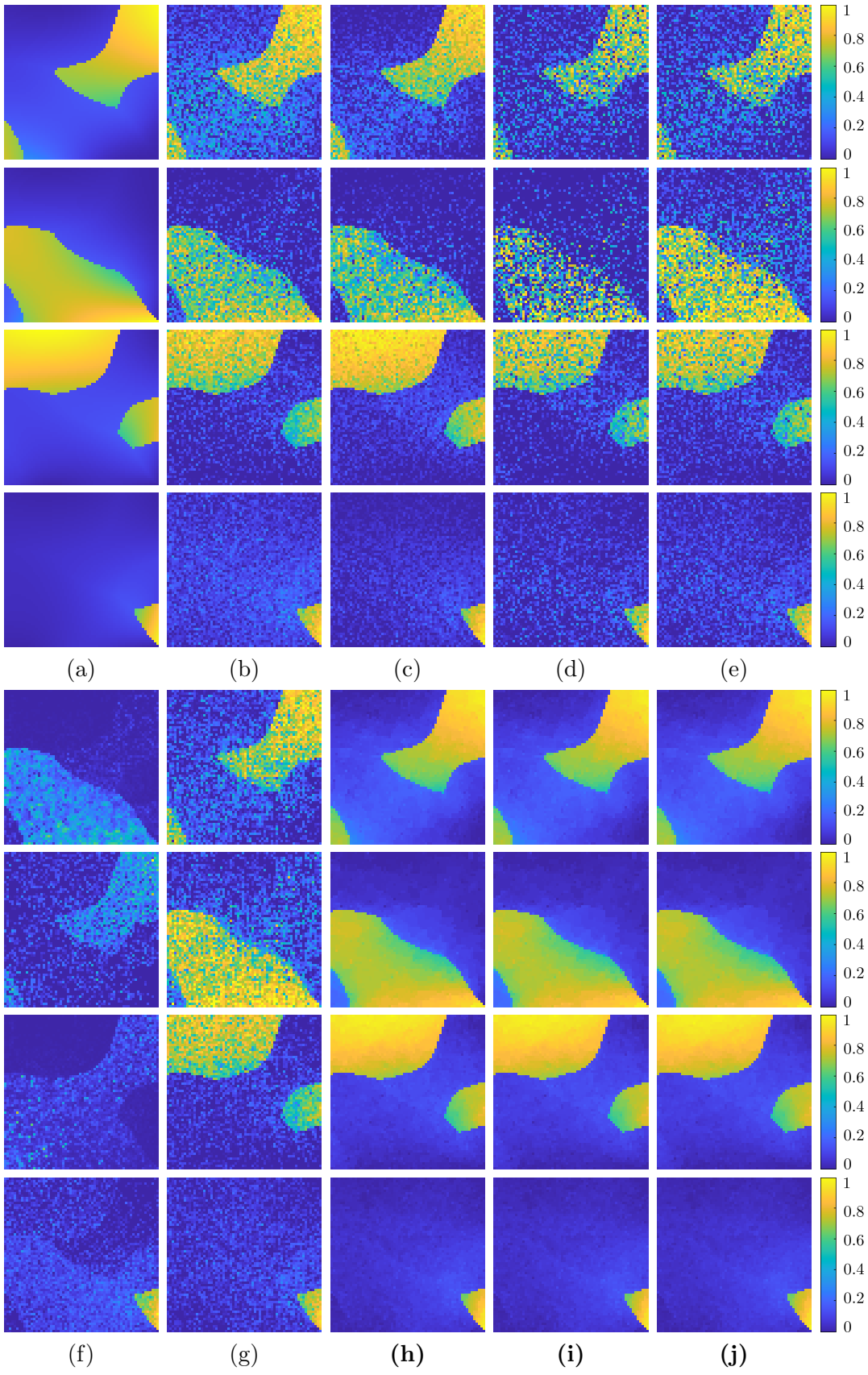


Figure 5.6: Unmixing results of abundance maps for the *Synth 1* experiments in Case 4. (a): Original abundance maps. (b): CLSUnSAL [87]. (c): JSTV [9]. (d): RSSUnTV [183]. (e): LGSU [163]. (f): UnDIP [151]. (g): MdLRR [194]. (h): **RHUIDR (HTV)**. (i): **RHUIDR (SSTV)**. (j): **RHUIDR (HSSTV)**.

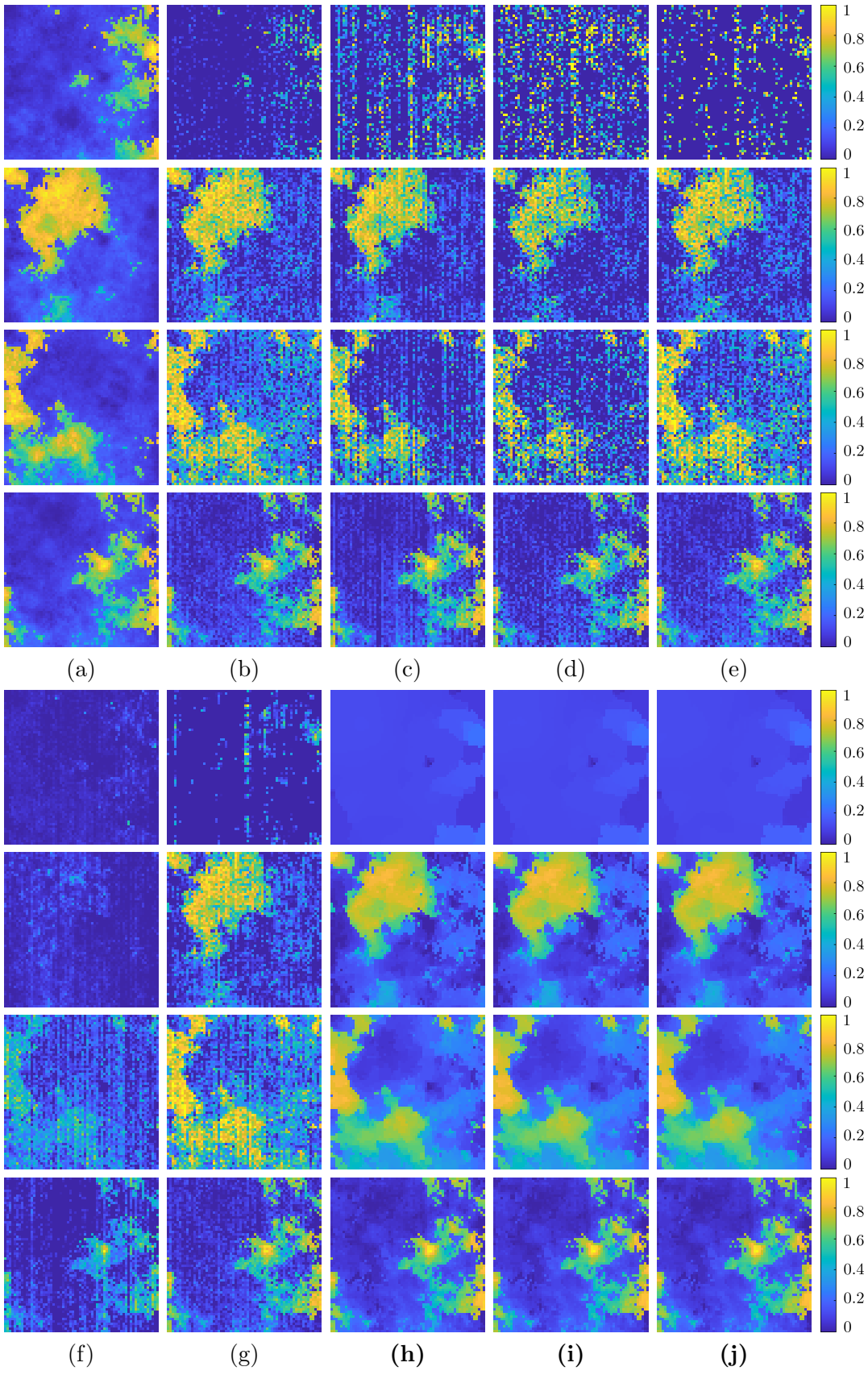


Figure 5.7: Unmixing results of abundance maps for the *Synth 2* experiments in Case 5. (a): Original abundance maps. (b): CLSUnSAL [87]. (c): JSTV [9]. (d): RSSUnTV [183]. (e): LGSU [163]. (f): UnDIP [151]. (g): MdLRR [194]. (h): **RHUIDR (HTV)**. (i): **RHUIDR (SSTV)**. (j): **RHUIDR (HSSTV)**.

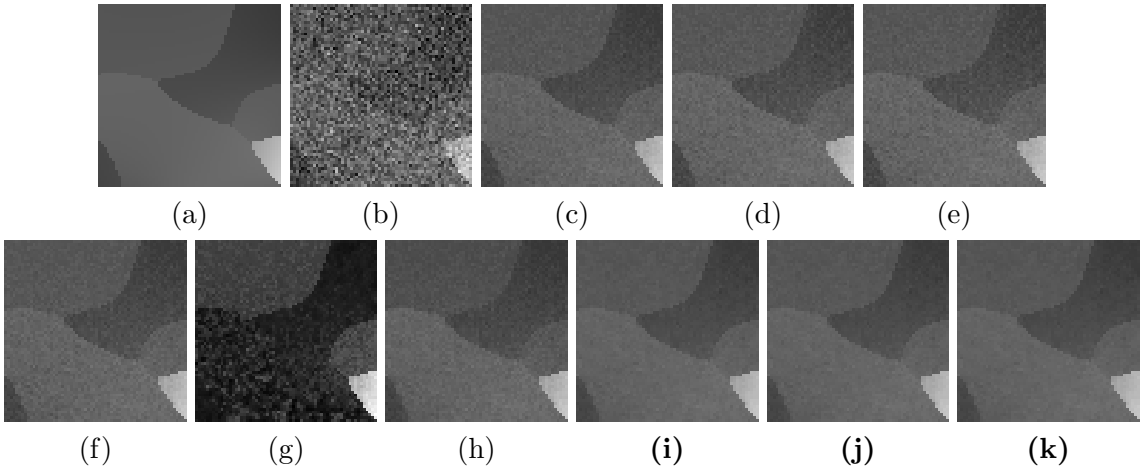


Figure 5.8: Reconstructed HS image results for the *Synth 1* experiments in Case 2. (a): Original HS image. (b): Noisy image. (c): CLSUnSAL [87]. (d): JSTV [9]. (e): RSSUnTV [183]. (f): LGSU [163]. (g): UnDIP [151]. (h): MdLRR [194]. (i): **RHUIDR (HTV)**. (j): **RHUIDR (SSTV)**. (k): **RHUIDR (HSSTV)**.

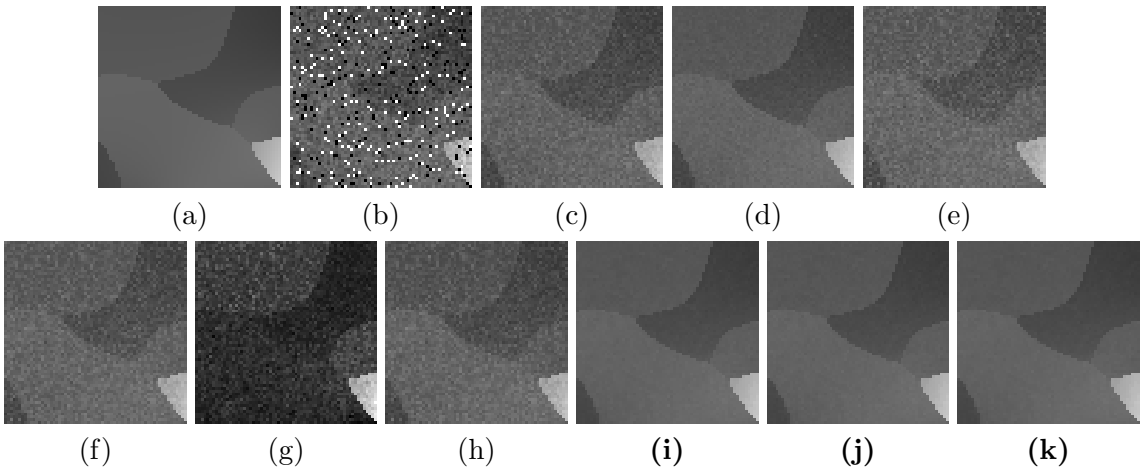
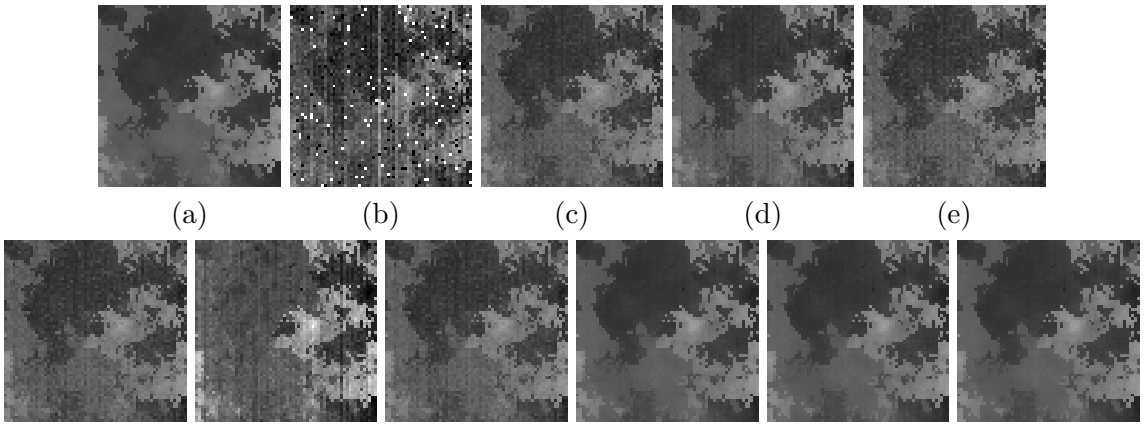


Figure 5.9: Reconstructed HS image results for the *Synth 1* experiments in Case 4. (a): Original HS image. (b): Noisy image. (c): CLSUnSAL [87]. (d): JSTV [9]. (e): RSSUnTV [183]. (f): LGSU [163]. (g): UnDIP [151]. (h): MdLRR [194]. (i): **RHUIDR (HTV)**. (j): **RHUIDR (SSTV)**. (k): **RHUIDR (HSSTV)**.



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

(f) (g) (h) **(i)** **(j)** **(k)**

Figure 5.10: Reconstructed HS image results for the *Synth 2* experiments in Case 5. (a): Original HS image. (b): Noisy image. (c): CLSUnSAL [87]. (d): JSTV [9]. (e): RSSUnTV [183]. (f): LGSU [163]. (g): UnDIP [151]. (h): MdLRR [194]. (i): **RHUIDR (HTV)**. (j): **RHUIDR (SSTV)**. (k): **RHUIDR (HSSTV)**.

5.4.4 Experiments With Real HS Images

Tables 5.7 and 5.8 show the SRE, RMSE, Ps, MPSNR, and MSSIM results for *Jasper Ridge* and *Samson*, respectively. The best and second-best results are highlighted in bold and underlined, respectively. CLSUnSAL, JSTV, RSSUn-TV, and UnDIP performed worse in all cases than RCHU. LGSU was the best for SRE, RMSE, and Ps in Cases 1 and 2 of the *Jasper Ridge* experiments. The reason for this is that LGSU accurately calculates the homogeneities of *Jasper Ridge* and the local sparsity promoted by the homogeneities significantly enhances the unmixing performance in these cases. In other cases, LGSU degraded the performance because it is not robust to sparse and stripe noise. In contrast, RHUIDR achieved the best SRE, RMSE, Ps, MPSNR, and MSSIM performances for the cases where the HS image is contaminated by noise that can be handled by the existing methods (Cases 1 and 2 for CLSUnSAL, RSSUn-TV, and LGSU, and Cases 1, 2, 3, and 4 for JSTV), except for Case 1, 2. In addition, RHUIDR can handle all three types of noise, resulting in the best performance for the other cases (Cases 5, 6). When comparing image-domain regularizations, HTV is better for *Jasper Ridge* and HSSTV is better for *Samson*.

Figure 5.11 shows the estimated abundance maps of *Jasper Ridge* in Case 2. Figure 5.12 depicts the estimated abundance maps of *Samson* in Case 6. As shown in Fig. 5.11 (b)-(g), if HS images are contaminated by Gaussian noise with a large standard deviation σ, all of the existing methods degraded their performance. Although CLSUnSAL, LGSU, and MdLRR achieved good SRE, RMSE, and MPSNR in Case 2 of the real data experiments, it yielded abundance maps with residual noise (see Fig. 5.11 (a), (e), and (g)). In Case 6, the abundance maps estimated by all the existing methods include residual noise, especially stripe noise (see Fig. 5.12, (b)-(g)). This indicates that they do not handle the stripe noise. In contrast, RHUIDR exactly estimated the abundance maps even under the conditions assumed by the existing methods, e.g., when the observed HS images are only contaminated by Gaussian noise (see Fig. 5.11). Furthermore, RHUIDR estimated the abundance maps by removing not only Gaussian and sparse noise but also stripe noise cleanly.

Figure 5.13 displays the reconstructed HS images of *Jasper Ridge* in Case 2. Figure 5.14 depicts the reconstructed HS images of *Samson* in Case 6. All of the existing methods resulted in noise remaining in the reconstructed HS images (see Figs. 5.13 and 5.14 (c)-(h)). In particular, they cannot handle stripe noise, and thus did not completely remove it in Case 6 (see Fig. 5.14 (c)-(h)). On the other hand, RHUIDR reconstructed the HS image cleanly (see Figs. 5.13 and 5.14 (i)-(k)). This verifies the effectiveness of the image-domain regularization.

Table 5.7: SRE, RMSE, Ps, MPSNR, and MSSIM in the Experiments Using *Jasper Ridge*.

Noise	Evaluation Target	Measures	Methods								
			CLSUnSAL [87]	JSTV [9]	RSSUn-TV [183]	LGSU [163]	UnDIP [151]	MdLRR [194]	**RHUIDR (HTV)**	**RHUIDR (SSTV)**	**RHUIDR (HSSTV)**
Case 1	Abundance	SRE	19.97	16.22	15.88	19.72	8.12	**20.23**	17.72	17.07	17.06
		RMSE	0.0431	0.0664	0.0690	0.0444	0.1688	**0.0412**	0.0559	0.0602	0.0603
		Ps	0.42	0.40	0.40	0.40	0.31	0.43	**0.44**	0.42	0.43
	Reconstructed HS image	MPSNR	37.67	36.80	36.71	**38.28**	25.37	37.85	37.35	37.02	36.96
		MSSIM	0.9858	0.9821	0.9817	0.9866	0.9129	0.9858	**0.9878**	0.9857	0.9864
Case 2	Abundance	SRE	15.24	11.11	10.95	12.92	5.64	12.44	**16.09**	15.59	15.62
		RMSE	0.0744	0.1197	0.1218	0.0972	0.2245	0.1011	**0.0674**	0.0715	0.0712
		Ps	0.37	0.34	0.34	0.34	0.31	0.36	**0.40**	**0.40**	**0.40**
	Reconstructed HS image	MPSNR	32.01	31.08	31.04	31.92	21.07	31.43	**33.40**	32.90	33.29
		MSSIM	0.9538	0.9409	0.9404	0.9477	0.8303	0.9438	**0.9687**	0.9629	0.9681
Case 3	Abundance	SRE	11.74	15.48	7.62	8.12	3.65	8.38	**17.49**	16.85	16.83
		RMSE	0.1112	0.0724	0.1789	0.1687	0.2823	0.1652	**0.0574**	0.0618	0.0619
		Ps	0.33	0.40	0.32	0.32	0.20	0.33	**0.44**	0.43	0.43
	Reconstructed HS image	MPSNR	25.44	35.70	24.85	25.05	16.74	24.89	**36.59**	36.26	36.23
		MSSIM	0.8701	0.9771	0.8442	0.8479	0.6797	0.8460	**0.9851**	0.9827	0.9837
Case 4	Abundance	SRE	8.83	14.38	4.83	4.98	3.84	5.61	**17.77**	17.28	17.27
		RMSE	0.1556	0.0821	0.2465	0.2424	0.2762	0.2357	**0.0556**	0.0588	0.0589
		Ps	0.28	0.38	0.27	0.27	0.19	0.28	**0.43**	0.42	0.42
	Reconstructed HS image	MPSNR	20.83	34.70	20.42	20.51	19.05	20.43	**36.30**	35.94	36.08
		MSSIM	0.7734	0.9716	0.7412	0.7435	0.7610	0.7429	**0.9830**	0.9804	0.9820
Case 5	Abundance	SRE	11.17	11.83	6.95	7.39	5.79	7.65	**16.37**	15.91	15.93
		RMSE	0.1188	0.1102	0.1932	0.1837	0.2207	0.1801	**0.0653**	0.0688	0.0687
		Ps	0.33	0.36	0.31	0.31	0.28	0.32	**0.42**	**0.42**	**0.42**
	Reconstructed HS image	MPSNR	25.09	32.19	24.47	24.65	20.87	24.50	**34.98**	34.51	34.86
		MSSIM	0.8613	0.9559	0.8347	0.8381	0.8001	0.8363	**0.9795**	0.9759	0.9788
Case 6	Abundance	SRE	10.60	9.36	6.36	6.62	3.28	6.99	**14.06**	13.56	13.59
		RMSE	0.1268	0.1463	0.2068	0.2006	0.2947	0.1955	**0.0852**	0.0902	0.0899
		Ps	0.31	0.32	0.29	0.29	0.18	0.30	**0.39**	**0.39**	**0.39**
	Reconstructed HS image	MPSNR	24.46	28.75	23.81	23.96	18.83	23.84	**31.45**	30.79	31.24
		MSSIM	0.8407	0.9102	0.8106	0.8136	0.7317	0.8115	**0.9566**	0.9459	0.9550

Table 5.8: SRE, RMSE, Ps, MPSNR, and MSSIM in the Experiments Using *Samson*.

Noise	Evaluation Target	Measures	Methods								
			CLSUnSAL [87]	JSTV [9]	RSSUn-TV [183]	LGSU [163]	UnDIP [151]	MdLRR [194]	**RHUIDR (HTV)**	**RHUIDR (SSTV)**	**RHUIDR (HSSTV)**
Case 1	Abundance	SRE	12.46	11.29	10.75	10.04	7.62	13.81	14.84	14.68	**15.11**
		RMSE	0.0998	0.1166	0.1241	0.1331	0.1909	0.0874	0.0761	0.0767	**0.0737**
		Ps	0.67	0.64	0.63	0.63	0.41	0.68	0.77	0.77	**0.78**
	Reconstructed HS image	MPSNR	35.55	35.06	34.91	35.17	22.39	35.73	37.85	37.23	**37.87**
		MSSIM	0.9772	0.9733	0.9722	0.9736	0.8705	0.9780	**0.9899**	0.9869	0.9897
Case 2	Abundance	SRE	7.87	6.63	6.35	6.41	2.21	7.31	10.22	9.98	**10.53**
		RMSE	0.1602	0.1973	0.2046	0.2024	0.3555	0.1836	0.1166	0.1185	**0.1128**
		Ps	0.57	0.52	0.51	0.51	0.22	0.54	**0.66**	0.65	**0.66**
	Reconstructed HS image	MPSNR	30.00	29.45	29.36	29.67	15.74	29.55	31.90	31.40	**32.03**
		MSSIM	0.9271	0.9144	0.9125	0.9165	0.7066	0.9166	0.9552	0.9484	**0.9569**
Case 3	Abundance	SRE	5.52	10.32	4.53	4.73	4.59	4.70	14.26	14.18	**14.63**
		RMSE	0.1964	0.1300	0.3022	0.2967	0.2703	0.2925	0.0804	0.0806	**0.0774**
		Ps	0.49	0.62	0.43	0.43	0.29	0.45	**0.76**	**0.76**	**0.76**
	Reconstructed HS image	MPSNR	23.14	33.91	22.66	22.83	16.30	22.51	36.78	36.16	**36.82**
		MSSIM	0.7912	0.9658	0.7677	0.7717	0.6529	0.7644	**0.9865**	0.9827	0.9864
Case 4	Abundance	SRE	3.17	9.17	3.31	3.44	1.76	3.34	13.39	13.22	**13.60**
		RMSE	0.2368	0.1464	0.3990	0.3986	0.3745	0.3905	0.0869	0.0877	**0.0847**
		Ps	0.41	0.59	0.35	0.35	0.24	0.36	**0.75**	0.74	**0.75**
	Reconstructed HS image	MPSNR	18.61	33.06	18.27	18.35	16.16	18.20	**35.85**	35.24	35.83
		MSSIM	0.6662	0.9593	0.6365	0.6380	0.5906	0.6351	**0.9824**	0.9783	0.9821
Case 5	Abundance	SRE	4.97	7.45	4.27	4.48	4.98	4.42	10.37	10.05	**10.75**
		RMSE	0.2065	0.1801	0.3096	0.3045	0.2584	0.2996	0.1144	0.1170	**0.1099**
		Ps	0.47	0.54	0.42	0.42	0.30	0.43	**0.69**	0.68	**0.69**
	Reconstructed HS image	MPSNR	22.81	30.52	22.35	22.50	20.83	22.21	33.70	33.11	**33.85**
		MSSIM	0.7822	0.9364	0.7587	0.7626	0.7465	0.7561	0.9725	0.9673	**0.9735**
Case 6	Abundance	SRE	4.55	5.36	3.87	4.02	3.63	3.98	9.76	9.31	**10.29**
		RMSE	0.2157	0.2433	0.3398	0.3369	0.3021	0.3319	0.1212	0.1257	**0.1156**
		Ps	0.47	0.48	0.40	0.40	0.25	0.41	0.69	0.68	**0.70**
	Reconstructed HS image	MPSNR	22.30	27.07	21.79	21.93	18.14	21.68	30.63	29.86	**30.83**
		MSSIM	0.7559	0.8700	0.7299	0.7333	0.6863	0.7277	0.9471	0.9330	**0.9500**

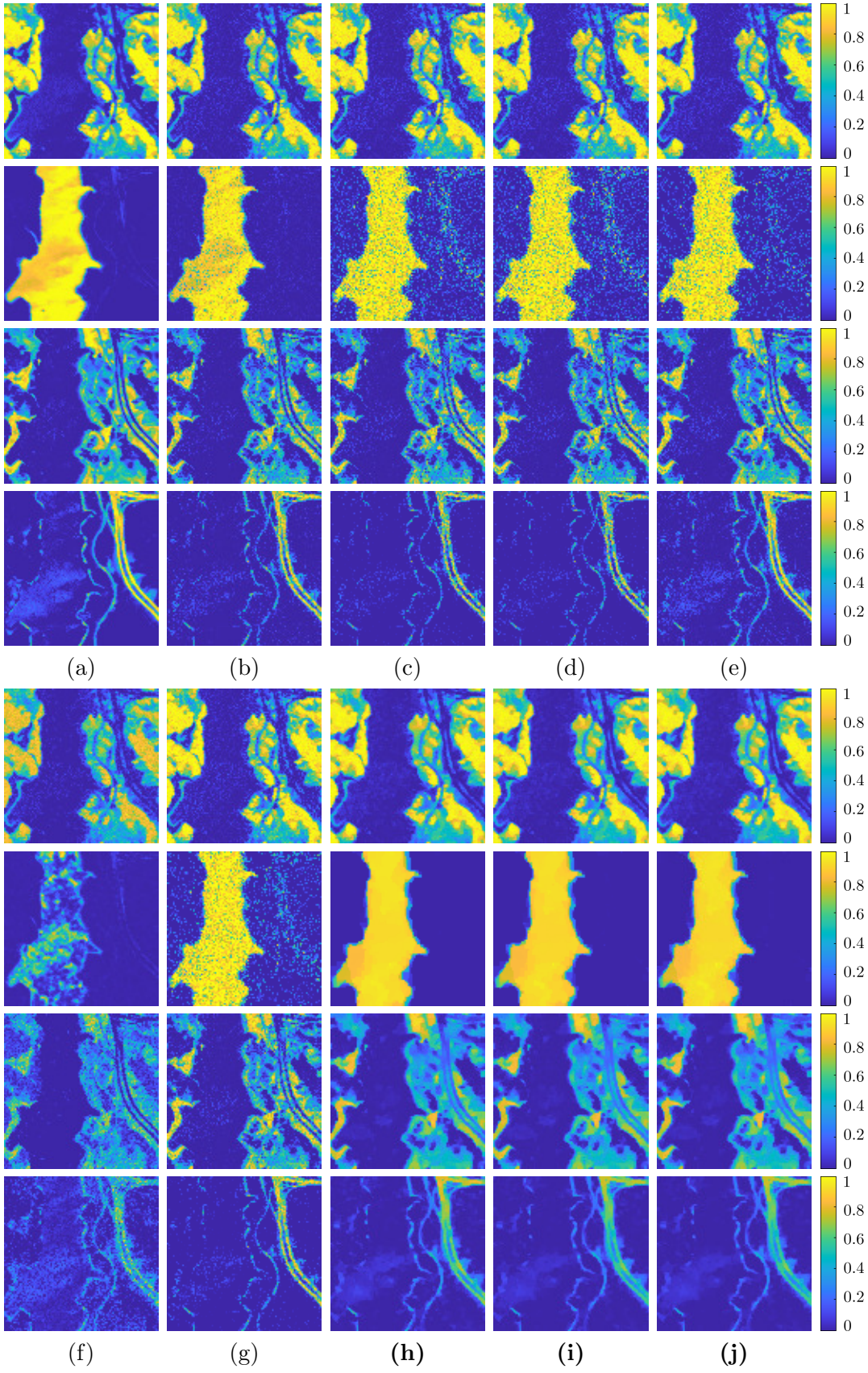


Figure 5.11: Unmixing results of abundance maps for the *Jasper Ridge* experiments in Case 2. (a): Original abundance maps. (b): CLSUnSAL [87]. (c): JSTV [9]. (d): RSSUnTV [183]. (e): LGSU [163]. (f): UnDIP [151]. (g): MdLRR [194]. (h): **RHUIDR (HTV)**. (i): **RHUIDR (SSTV)**. (j): **RHUIDR (HSSTV)**.

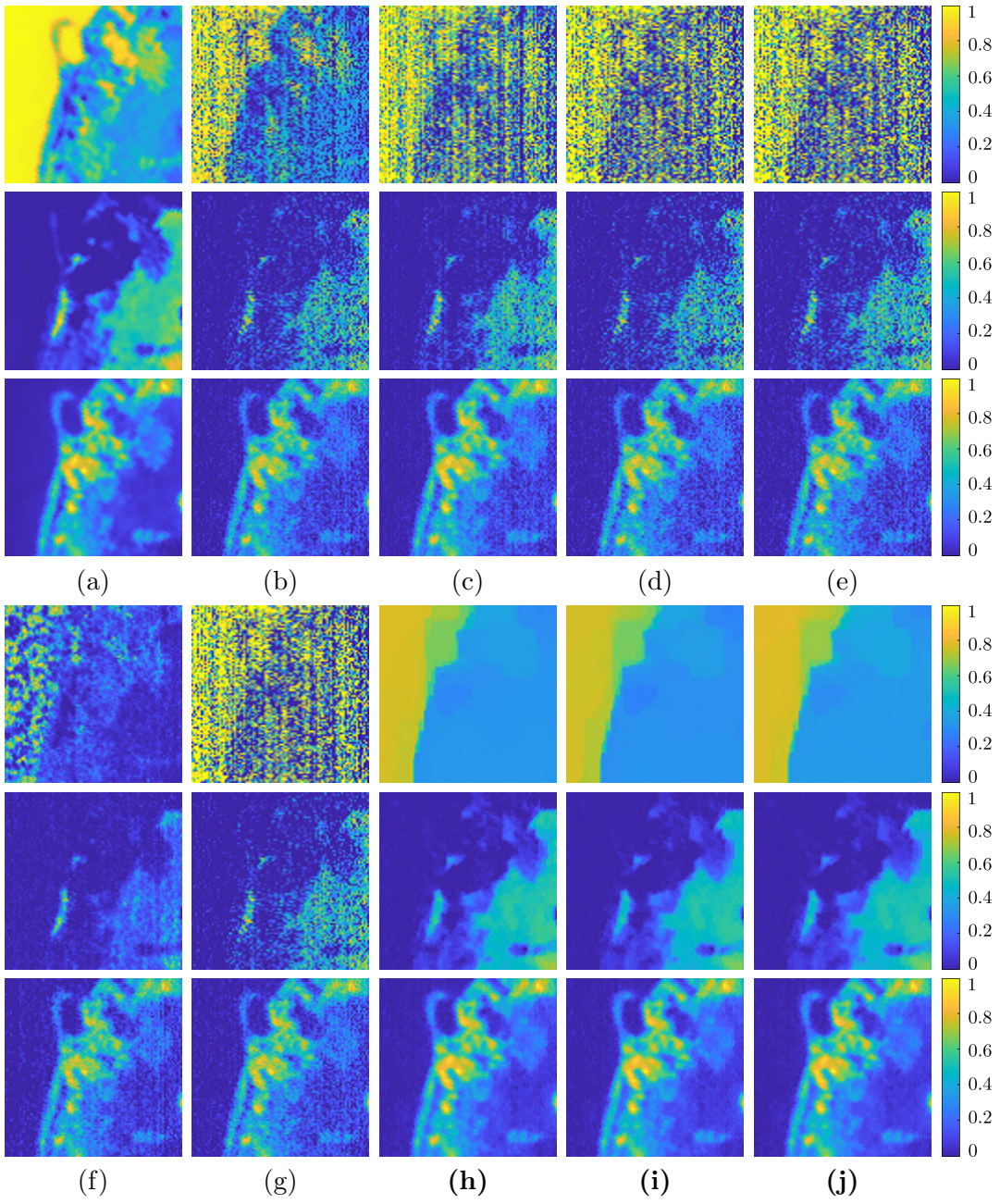


Figure 5.12: Unmixing results of abundance maps for the *Samson* experiments in Case 6. (a): Original abundance maps. (b): CLSUnSAL [87]. (c): JSTV [9]. (d): RSSUnTV [183]. (e): LGSU [163]. (f): UnDIP [151]. (g): MdLRR [194]. (h): **RHUIDR (HTV)**. (i): **RHUIDR (SSTV)**. (j): **RHUIDR (HSSTV)**.

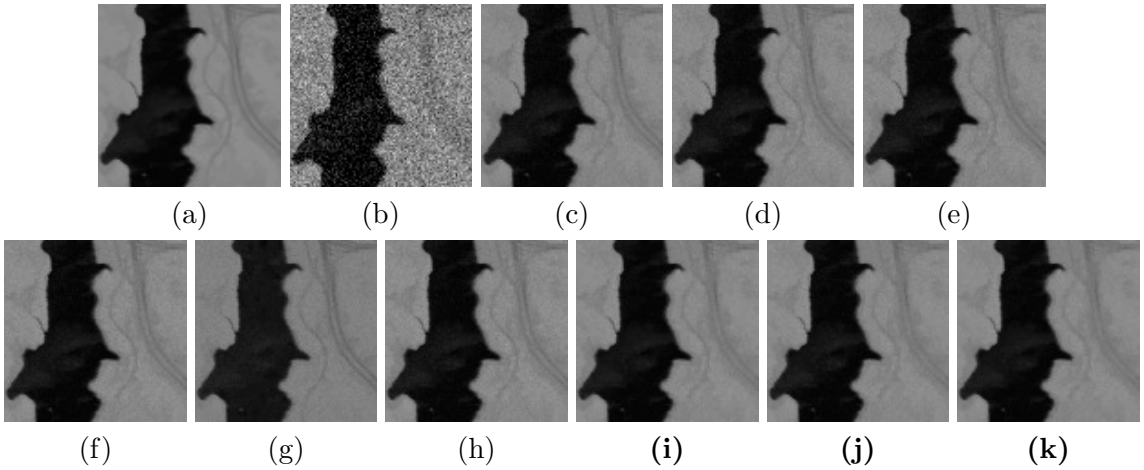


Figure 5.13: Reconstructed HS image results for the *Jasper Ridge* experiments in Case 2. (a): Original HS image. (b): Noisy image. (c): CLSUnSAL [87]. (d): JSTV [9]. (e): RSSUn-TV [183]. (f): LGSU [163]. (g): UnDIP [151]. (h): MdLRR [194]. (i): **RHUIDR (HTV)**. (j): **RHUIDR (SSTV)**. (k): **RHUIDR (HSSTV)**.

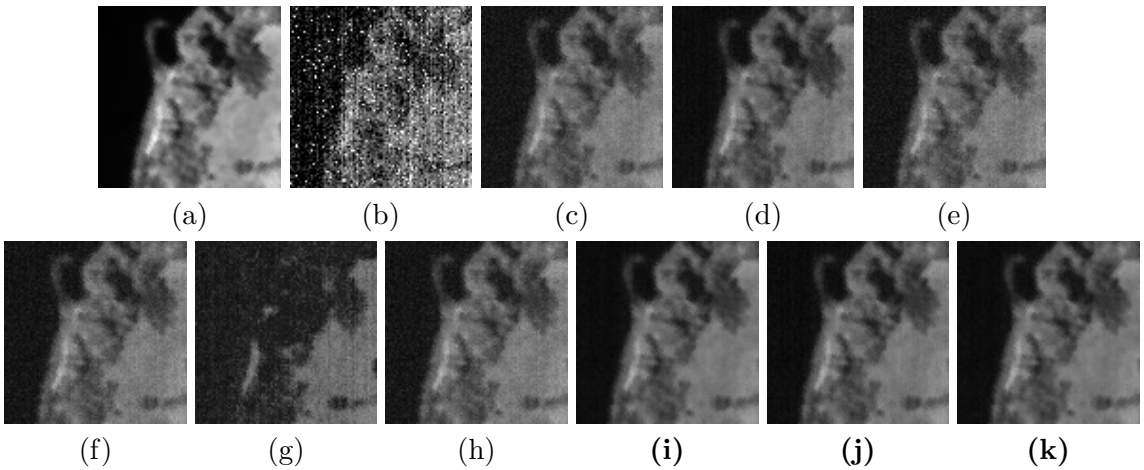


Figure 5.14: Reconstructed HS image results for the *samson* experiments in Case 6. (a): Original HS image. (b): Noisy image. (c): CLSUnSAL [87]. (d): JSTV [9]. (e): RSSUn-TV [183]. (f): LGSU [163]. (g): UnDIP [151]. (h): MdLRR [194]. (i): **RHUIDR (HTV)**. (j): **RHUIDR (SSTV)**. (k): **RHUIDR (HSSTV)**.

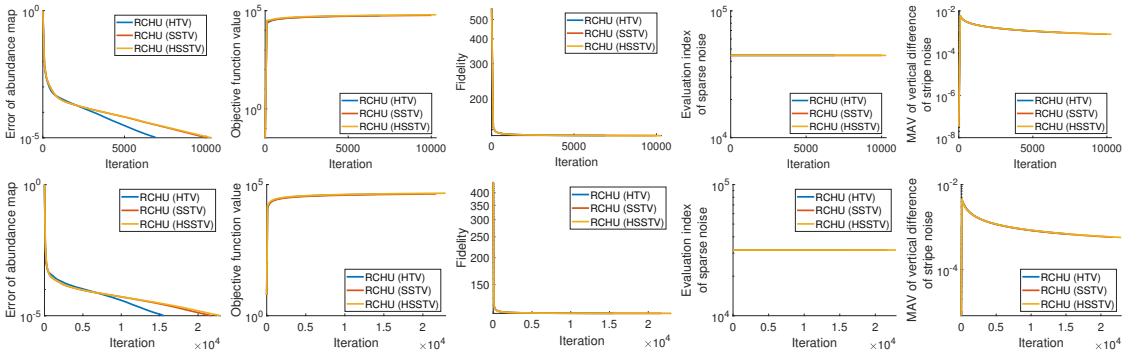


Figure 5.15: Convergence analysis using the real images. The top row shows the results of experiments using *Jasper Ridge*. The bottom row shows the results of experiments using *Samson*. (a): The relative error of abundance maps $\|\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}\|_F / \|\mathbf{A}^{(t+1)}\|_F$ versus iteration t. (b): Objective function value $\|\mathbf{A}^{(t)}\|_{1,2,r} + \lambda_1 \|\mathbf{D}(\mathbf{A}^{(t)})\|_1 + \lambda_2 \mathcal{R}(\mathfrak{K}(\mathbf{E}\mathbf{A}^{(t)})) + \lambda_3 \|\mathbf{L}^{(t)}\|_1$ versus iteration t. (c): The ℓ_2 distance between $\mathbf{V}$ and $\mathbf{E}\mathbf{A}^{(t)} + \mathbf{S}^{(t)} + \mathbf{L}^{(t)}$ versus iteration t. (d): The ℓ_1 norm of $\mathbf{S}^{(t)}$ versus iteration t. (e): The mean absolute values (MAV) of $\mathbf{D}_v(\mathbf{L}^{(t)})$ versus iteration t.

5.4.5 Computational Cost and Convergence Analysis

We measured the actual running times using MATLAB (R2022b) on a Windows 11 computer with an Intel Core i9-13900 1.0GHz processor, 32GB of RAM, and NVIDIA GeForce RTX 4090. For *Jasper Ridge* experiments in Case 6, RHUIDR using HTV, SSTV, and HSSTV took 169.76, 153.95, and 161.86 of actual running times [s], respectively. For *Samson* experiments in Case 6, RHUIDR using HTV, SSTV, and HSSTV took 159.60, 144.48, and 158.76 of actual running times [s], respectively.

In addition, we experimentally analyzed the convergence of our method. Figure 5.15 plots the relative error of abundance maps: $\|\mathbf{A}^{(t+1)} - \mathbf{A}^{(t)}\|_F / \|\mathbf{A}^{(t)}\|_F$, the objective function values, the Frobenius distance between $\mathbf{V}$ and $\mathbf{E}\mathbf{A}^{(t)} + \mathbf{S}^{(t)} + \mathbf{L}^{(t)}$, the ℓ_1 norm of $\mathbf{S}^{(t)}$, and the mean absolute values (MAV) of $\mathfrak{D}_v(\mathbf{L}^{(t)})$ for *Jasper Ridge* experiments and *Samson* experiments. The relative error of abundance maps decreased (Fig. 5.15 (a)). While getting larger as the number of iterations increases, the objective function value asymptotically approaches a certain value (Fig. 5.15 (b)). This is often found when solving optimization problems involving hard constraints, such as a data fidelity constraint $\mathbf{E}\mathbf{A} + \mathbf{S} + \mathbf{L} \in B^{\mathbf{V}}_{F,\varepsilon}$, a sparsity constraint $\mathbf{S} \in B_{1,\eta}$, and a flatness constraint $\mathfrak{D}_v(\mathbf{L}) = \mathbf{O}$. The ℓ_2 distance and the MAV become smaller, where we can see that the variables are updated by P-PDS to approach the solution of our constrained convex optimization.

5.4.6 Ablation Experiments

To demonstrate the effectiveness of the image-domain regularization (the third term of Eq. (5.10)), we compared RHUIDR performance with the performance when the image-domain regularization was removed (referred to as RHUIDR (–)). This ablation experiment was performed by using four images (*Synth 1*, *Synth 2*, *Jasper Ridge*, and *Samson*) for Case 6.

Table 5.9 shows the SRE, RMSE, Ps, MPSNR, and MSSIM results for the ablation experiment. The best and second-best results are highlighted in bold and underlined, respectively. RCHU with the image-domain regularization was superior to RCHU without the image-domain regularization. In particular, the image-domain regularization contributed to an improvement in SRE, RMSE, and Ps and a significant improvement in MPSNR and MSSIM. This implies that the reconstructed HS image has the desirable spatio-spectral

Table 5.9: RSE, RMSE, Ps, MPSNR, and MSSIM of Ablation Experiments.

Image	Evaluation Target	Measures	Methods			
			RHUIDR –	**RHUIDR (HTV)**	**RHUIDR (SSTV)**	**RHUIDR (HSSTV)**
Synth 1	Abundance	SRE	19.88	**20.13**	19.72	19.63
		RMSE	0.0390	**0.0377**	0.0393	0.0396
		Ps	0.76	0.79	**0.80**	0.82
	Reconstructed HS image	MPSNR	36.21	38.85	36.95	**39.44**
		MSSIM	0.9420	0.9702	0.9527	**0.9753**
Synth 2	Abundance	SRE	6.83	7.94	**7.97**	7.92
		RMSE	0.1464	0.1464	**0.1459**	0.1468
		Ps	0.67	0.68	**0.69**	0.68
	Reconstructed HS image	MPSNR	34.92	35.97	35.09	**36.11**
		MSSIM	0.9626	0.9735	0.9641	**0.9737**
Jasper Ridge	Abundance	SRE	13.42	**14.06**	13.56	13.59
		RMSE	**0.0843**	0.0852	0.0902	0.0899
		Ps	**0.39**	**0.39**	**0.39**	**0.39**
	Reconstructed HS image	MPSNR	30.86	**31.45**	30.79	31.24
		MSSIM	0.9450	**0.9566**	0.9459	0.9550
Samson	Abundance	SRE	9.13	9.76	9.31	**10.29**
		RMSE	0.1280	0.1212	0.1257	**0.1156**
		Ps	0.67	0.69	0.68	**0.70**
	Reconstructed HS image	MPSNR	29.49	30.63	29.86	**30.83**
		MSSIM	0.9253	0.9471	0.9330	**0.9500**

property, resulting in the estimation of more appropriate abundance maps.

5.4.7 Summary

We summarize the experimental discussion as follows.

- From the results of experiments in Case 1, Case 2, Case 3, and Case 4, we see that image-domain regularizations improve the unmixing performance.
- The results of experiments in Case 5 and Case 6 verify that RHUIDR accurately estimates abundance maps if HS images are degraded by various types of noise.
- RHUIDR achieves good unmixing performance in experiments on both synthetic and real HS images.

5.5 Concluding Remarks

In this chapter, we have proposed a new method for noise-robust unmixing. RHUIDR adopts the image-domain regularization and explicitly models three types of noises. We have formulated the unmixing problem as a constrained convex optimization problem that includes the regularization, and have developed the optimization algorithm based on P-PDS. Experiments on synthetic and real HS images have demonstrated the superiority of RHUIDR over existing methods. RHUIDR will have strong impacts on the field of remote sensing, including the estimation of abundance maps from HS images taken in measurement environments with severe degradation. For future work, we achieve more noise-robust blind unmixing by incorporating RHUIDR into learning-based methods.

Chapter 6

General Conclusion

This dissertation has studied convex optimization techniques for remote sensing analysis with the ultimate goal of establishing a unified framework that can handle a wide range of all target data and tasks. Although there is still a long way to go to achieve the ultimate goal, we have addressed the following research question: Can we exploit degradation models, statistical information, and problem structures to flexibly determine appropriate values of parameters in remote sensing data analysis? Specifically, we have adopted constraint modeling to eliminate the interdependence of regularization parameters and developed a method that employs the structure of an optimization problem to resolve the difficulty of tuning the stepsizes of its optimization algorithm.

In Chapter 3, we designed a general framework that can handle various types of target data and a novel characterization of stripe noise. These enabled us to establish a method to consistently remove stripe noise without changing parameter settings for any target data. Through experiments with a comprehensive comparison of various image regularizations and stripe noise characterizations for HS images and IR videos, we have shown that the framework performed several dB better in PSNR than frameworks using existing characterizations for both data.

Chapter 4 has proposed a method for determining the stepsizes of P-PDS, an efficient convex optimization algorithm. Specifically, by introducing the operator norms of the linear operators included in a target optimization problem, our method reflects the problem structure in determining the appropriate stepsizes such that the convergence of P-PDS is fast for any optimization problem. The operator norms of linear operators present in remote sensing data analysis are often known, and thus the convenience of our method in real-world applications is not compromised. In addition, a variable-wise diagonal preconditioning approach does not increase the complexity of the operations in the optimization problem, thus significantly reducing the execution time. Through experiments of applying the approach to two remote sensing data analysis tasks (hyperspectral image mixture denoising and mixel decomposition) and one signal analysis task (graph signal restoration), we have confirmed that our method is more convenient and several to several hundred times faster than existing step-sizing methods.

Chapter 5 has established a noise-robust unmixing method by taking advantage of our methods proposed in Chapters 3 and 4. In addition to regularizations for an abundance matrix, we have designed an optimization problem by introducing regularizations for the HS image reconstructed from estimated abundance maps and endmembers, and constraint modeling including flatness constraints proposed in Chapter 3. These allow us to deal with high-level and diverse noise while facilitating the setting of regularization parameters. To solve the optimization problem, we have developed a P-PDS-based algorithm that does not require stepsize adjustment using the stepsize determination method proposed in Chapter 4. Experiments on HS images with diverse noise have demonstrated that our

method yielded several decibel-better SREs than existing methods, especially in situations of severe noise.

6.1 Limitations

As described in the previous section, this dissertation contributes to the establishment of convex optimization techniques for flexible and reliable remote sensing data analysis. However, this research still has the following several limitations.

6.1.1 Modeling of Regularizations as Constraints

In Chapters 3, 4, and 5, we have designed optimization problems using constraint modeling mainly for degradation characterizations. Since this constraint modeling approach can improve the flexibility and reliability of remote sensing data analysis, it is better to handle regularization using constraint modeling. However, achieving this requires further development of signal processing and optimization techniques. In fact, in the optimization problem in Chapter 5, regularizations for reconstructed images were included in the objective function due to the following two specific difficulties.

First, optimization algorithms may not always be constructed by realistic and efficient computations when adopting constraint modeling. The characterization of degradation is accomplished using an ℓ_2 (Frobenius) norm and an ℓ_1 norm. Fortunately, the constraints using these functions (ℓ_2 ball and ℓ_1 ball) are easy to handle in the optimization process because the metric projections onto an ℓ_2 ball and an ℓ_1 ball are efficiently computed. However, not all functions have this property, e.g., if we use a mixed $\ell_{1,2}$ norm (which appeared numerous times in this dissertation) for constraint modeling, it is difficult to construct an optimization algorithm because the projection onto a mixed $\ell_{1,2}$ ball is not easy to compute. Therefore, to introduce regularized functions, it is essential to establish optimization techniques that can efficiently handle any functions for constraint modeling.

Second, for parameters associated with regularization constraints, determining appropriate values is difficult. For parameters associated with degradation characterization constraints, this dissertation has determined their appropriate values based on the statistical information of the degradations. For example, the parameter of a data-fidelity constraint can be determined using the standard deviation of Gaussian noise, and the parameter of a sparse constraint can be set using the rate of sparse noise. However, it is difficult to assume the statistical or mathematical properties behind remote sensing data, which prevents us from setting parameters consistently. To this end, the discovery of helpful measurement information and the development of signal processing techniques are necessary to establish such a methodology.

6.1.2 Capabilities for Modeling Prior Knowledges

This dissertation has developed techniques for convex optimization, which is a powerful tool for obtaining stable results while incorporating prior knowledge into analysis. However, convex optimization has the following limitations concerning the accuracy and capability of modeling prior knowledge.

First, convex optimization sometimes fails in rigorous models of sparsity and low-rankness, which are often required in remote sensing data analysis, and thus degrades the performance of the analysis. As an example of sparsity, our method in Chapter 5 has modeled the group sparsity of an abundance matrix by mixed $\ell_{1,2}$ norms. This function does not sufficiently promote sparsity, resulting in the degradation of the unmixing performance in some cases (see in particular Figure 5.12).

Then, convex optimization techniques cannot deal with situations where the operation representing/converting data from/to the sparse or low-rank component cannot be fully modeled by hand. In remote sensing data analysis, there are many situations in which such conversions are not completely known, for example, sparse representation [15, 23, 67, 191, 192], which captures similar structures non-local when the operator for sparse from/to conversion is not known, and blind unmixing [63, 64, 106], which is performed under the assumption that some or all of the constituent dictionaries are unknown. Such analysis tasks are often not possible as convex optimization problems.

Addressing the above issue requires techniques beyond convex optimization techniques. In other words, the introduction of nonconvex optimization techniques is crucial for the further development of remote sensing data analysis.

6.1.3 Range of Applications

The ultimate goal of this research is to establish a framework that can handle a wide range of all target data and tasks. As part of this process, we have demonstrated its effectiveness in three analysis tasks: mixed denoising (including destriping) of HS images, mixel of HS images, and IR video restoration. However, due to the diversity of measurement and analysis processes, there are other remote sensing data analysis tasks, such as anomaly detection in hyperspectral images, spatio-temporal fusion of HS-MS images, speckle removal in aperture radar images, and motion detection and seismic data recovery in synthetic aperture radar images. Demonstrating the effectiveness of the results of this research in these analysis tasks is also necessary to achieve the ultimate goal.

6.2 Future works

6.2.1 Establishment of Optimization Techniques for Modeling Regularization as Constraints

As mentioned in Section 6.1.1, in order to model regularization as constraints, two techniques need to be developed: techniques for efficiently solving optimization problems, including constraint modeling of any functions, and techniques for estimating their appropriate parameters. For the former, we will consider developing epigraphical projections [47,94,101,139], which decompose mixed norms into norms that can be easily handled as constraints. For the latter, we will try out a wide range of techniques, from the simple method of pre-denoising observed data and estimating its appropriate function value to advanced estimation methods, such as the L-Curve [62, 72, 130] and the Stein unbiased risk estimate [5, 54, 169].

6.2.2 Introduction of Modeling Beyond Convex Optimization

Introducing nonconvex optimization techniques can overcome the limitations of the accuracy and capability of modeling prior knowledge. Among many nonconvex optimization techniques, a class of optimization techniques that can guarantee the convergence of optimization algorithms is crucial for obtaining stable analytical results. As such nonconvex optimization techniques, we are focusing on block multiconvex [198] and difference-of-convex (DC) optimization [12, 168], which can construct algorithms that can guarantee convergence.

Block multiconvex optimization handles optimization problems involving products of variables in convex functions. By formulating the analysis tasks as such optimization problems, a matrix or tensor can be decomposed into multiple vectors, matrices, or tensors,

and thus block multiconvex optimization is used for sparse representation-based analysis [15,23,67,191,192] and blind unmixing [63,64,106]. By re-modeling prior knowledge as constraints rather than convex functions, we can develop remote sensing data analysis that captures more a priori information while improving flexibility and reliability. In addition, when solving block multiconvex optimization problems, many methods take a strategy that alternately solves a convex optimization subproblem while fixing some variables. Incorporating the results of Chapter 4 into solving this convex optimization subproblem improves the convenience of block multiconvex optimization.

DC optimization minimizes a function created by subtracting a convex function from a convex function. This technique has attracted attention because of its ability to promote sparsity and low-rankness more strongly than convex optimization and thus can develop various remote sensing data analysis tasks. In fact, DC optimization is beginning to be used in remote sensing data analysis [93, 210]. In addition, optimization algorithms for solving DC optimization problems are constructed using several convex optimization techniques. In short, convex optimization is a fundamental technique for DC optimization. Therefore, the techniques established by this dissertation can promote the convenience of DC optimization.

6.2.3 Extension of A Framework to Handle A Wider Range of Analysis Tasks

This dissertation has not dealt with various types of remote sensing data analysis. In those analyses, optimization-based methods have been proposed, such as for anomaly detection in HS images [93,107,112,160], for spatiotemporal fusion of remote sensing images [89,90], for despeckling of SAR images [19, 172], for motion detection in SAR images [83], for seismic data recovery [15, 44], and for time-series image analysis [25, 26, 128, 133, 134]. Developing them by taking advantage of the results of this study, in future work, we will reach the ultimate goal of establishing a remote sensing data analysis framework for a wider range of analysis tasks. In addition, as proposed in Chapter 5, we will create an environment to use the knowledge found in the area of analysis tasks can across domains.

6.3 Closing Remarks

We have discussed the limitations of this study and clarified the directions for future works for the development of remote sensing data analysis. Finally, this dissertation concludes with a discussion of the implications of this research.

In this paper, we have addressed the fundamental issues lacking in the signal processing and optimization fields while scrutinizing the demands in the field of remote sensing data analysis in order to build optimization techniques for remote sensing data analysis. This results in their mutual development, and thus this research has significantly contributed to those fields. In addition, it is a milestone in the completion of analysis to obtain important information from data that is severely degraded due to extreme measurement environment factors and hardware factors such as performance and defects of measurement equipment. Therefore, from the perspective of overcoming limitations of the measurement environment and hardware, the significance of this research is significant.

This research establishes convex optimization techniques that are anticipated to remain relevant for the foreseeable future, even amidst ongoing advancements in hardware. The versatility of these techniques becomes particularly crucial in scenarios involving significant data degradation. This degradation often arises due to the evolving nature of measurement equipment and methods, which are influenced by various factors including

cost, performance, and the specific needs dictated by different environments and objectives. Let us consider the following situations.

- Take, for instance, stripe noise, a phenomenon addressed in this study. Stripe noise typically originates from line-scan measurement methods [117], such as whiskbroom and pushbroom techniques. Although alternative measurement methods exist, each comes with its own set of advantages and drawbacks, making them suitable for different applications [71,193]. Consequently, line-scan methods are expected to remain in use, underscoring the importance of technology that can effectively mitigate inherent stripe noise.

- Furthermore, remote sensing data acquisition often requires measurements under extremely poor conditions, such as low illumination. Even under such harsh conditions, a trade-off between measurement performance and cost is inevitable. As a result, the acquired data is often contaminated with severe degradation. Techniques for analyzing the components of such severely degraded data (e.g., the noise-robust unmixing proposed in Chapter 5) will remain indispensable.

Therefore, the foundational work of this research in developing such technologies will maintain its significance well into the future.

Appendix A

Proof of Lemma 4.3.1

Proof. Let r be the rank of $\mathbf{A}$ and $\sigma_1(\mathbf{A}), \ldots, \sigma_r(\mathbf{A})$ be the singular values of $\mathbf{A}$. Then, $\mathbf{A}$ can be decomposed as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*, \tag{A.1}$$

where $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{n\times r}$ satisfy $\mathbf{U}^*\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^*\mathbf{V} = \mathbf{I}$. Then, we introduce an $r \times r$ unitary matrix $\mathbf{W}$ and define $\mathbf{B}$ and $\mathbf{C}$ as

$$\mathbf{B} = \mathbf{U}\mathbf{\Sigma}^{1-\beta}\mathbf{W}^*, \mathbf{C} = \mathbf{W}\mathbf{\Sigma}^{\beta}\mathbf{V}^*, \tag{A.2}$$

where $\mathbf{\Sigma}^{1-\beta} = \mathrm{diag}(\sigma_1(\mathbf{A})^{1-\beta}, \ldots, \sigma_r(\mathbf{A})^{1-\beta})$ and $\mathbf{\Sigma}^{\beta} = \mathrm{diag}(\sigma_1(\mathbf{A})^{\beta}, \ldots, \sigma_r(\mathbf{A})^{\beta})$. It is clear that $\mathbf{A} = \mathbf{B}\mathbf{C}$. In turn, we obtain from the definition that

$$\begin{aligned} \|\mathbf{B}\mathbf{x}\|_2^2 &= \|\mathbf{\Sigma}^{1-\beta}\mathbf{W}^*\mathbf{x}\|_2^2 \\ &\leq \sigma_1(\mathbf{A})^{2-2\beta}\|\mathbf{W}^*\mathbf{x}\|_2^2 \\ &= \sigma_1(\mathbf{A})^{2-2\beta}\|\mathbf{x}\|_2^2. \end{aligned} \tag{A.3}$$

Hence

$$\|\mathbf{B}\|_{\mathrm{op}} = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|\mathbf{B}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sigma_1(\mathbf{A})^{1-\beta}. \tag{A.4}$$

Arguing similarly, $\mathbf{C}$ satisfies $\|\mathbf{C}\|_{\mathrm{op}} = \sigma_1(\mathbf{A})^{\beta}$.

□

Acknowledgment

This dissertation is the end of my long journey to obtain a Ph.D. degree. I would like to take the opportunity to express my heartfelt gratitude here to the people to whom I owe for this work.

I would like to express my deep and sincere gratitude to my advisor, Associate Professor Shunsuke Ono. He has taught me a wide range of things from basic research skills to the importance of having the spirit to create new values. Without his teachings, I would not have been able to grow.

Next, I would like to express my deepest gratitude to my colleague Eisuke Yamagata, Tokyo Institute of Technology, Japan. I believe that through our friendly competition, we have been able to improve each other. He has been indispensable in the writing of this dissertation, and will continue to be so in the future.

I would also like to thank Assistant Professor Saori Takeyama, Tokyo Institute of Technology, Japan, for her support and advice to improve the quality of experiments.

Then, I am grateful to Shino Ogata for her invaluable help with all the administrative tasks at Tokyo Institute of Technology.

I am also grateful to all the members of my laboratory, for their warm friendship.

In addition, I would like to express my deep gratitude to Professor Hideaki Ishii, Professor Tsuyoshi Murata, Professor Isao Ono, and Professor Rio Yokota, Tokyo Institute of Technology, Japan, for serving as members of the examining committee of this dissertation.

Of course, I am indebted to my relatives and friends for their support in enriching my doctoral life.

This work was supported by Grant-in-Aid for the Japan Society for the Promotion of Science (JSPS) Fellows (DC2) and Japan Science and Technology Agency (JST) ACT-X.

References

[1] AVIRIS. https://aviris.jpl.nasa.gov/data/free_data.html.

[2] BU-TIV Dataset. http://csr.bu.edu/BU-TIV/BUTIV.html.

[3] GIC. http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

[4] SpecTIR. http://www.spectir.com/free-data-samples/.

[5] The projected GSURE for automatic parameter tuning in iterative shrinkage methods. *Appl. Comput. Harmon. Anal.*, 30(3):407–422, 2011.

[6] H. Aetesam, K. Poonam, and S. K. Maji. Proximal approach to denoising hyperspectral images under mixed-noise model. *IET Image Process.*, 14(14):3366–3372.

[7] M. Afonso, J.M. Bioucas-Dias, and M. Figueiredo. An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Trans. Image Process.*, 20(3):681–695, 2011.

[8] H. K. Aggarwal and A. Majumdar. Hyperspectral image denoising using spatio-spectral total variation. *IEEE Geosci. Remote Sens. Lett.*, 13(3):442–446, Mar. 2016.

[9] H. K. Aggarwal and A. Majumdar. Hyperspectral unmixing in the presence of mixed noise using joint-sparsity and total variation. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 9(9):4257–4266, Sep. 2016.

[10] N. Audebert, B. Le Saux, and S. Lefevre. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geosci. Remote Sens. Mag.*, 7(2):159–173, 2019.

[11] G. Baier, W. He, and N. Yokoya. Robust nonlocal low-rank sar time series despeckling considering speckle correlation by total variation regularization. *IEEE Trans. Geosci. Remote Sens.*, 58(11):7942–7954, 2020.

[12] S. Banert and R. I. Boţ. A general double-proximal gradient algorithm for d.c. programming. *Math. Program.*, 178(1-2):301–326, 2019.

[13] A. Beck and Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.*, 2(1):183–202, 2009.

[14] S. Becker, J. Fadili, and P. Ochs. On quasi-Newton forward-backward splitting: Proximal calculus and convergence. *SIAM J. Optim.*, 29(4):2445–2481, 2019.

[15] Simon Beckouche and Jianwei Ma. Simultaneous dictionary learning and denoising for seismic data. *Geophysics*, 79(3):A27–A31, 2014.

[16] J. W. Beletic, R. Blank, D. Gulbransen, D. Lee, M. Loose, E. C. Piquette, T. Sprafke, W. E. Tennant, M. Zandian, and J. Zino. Teledyne Imaging Sensors: infrared imaging technologies for astronomy and civil space. In *Proc. SPIE*, volume 7021, pages 70210H–1–14, Jul. 2008.

[17] P. Berger, G. Hannak, and G. Matz. Graph signal recovery via primal-dual algorithms for total variation minimization. *IEEE J. Sel. Topics Signal Process.*, 11(6):842–855, Sep. 2017.

[18] R. Bhatia and F. Kittaneh. Norm inequalities for partitioned operators and an application. *Math. Ann.*, 287(4):719–726, 1990.

[19] J. M. Bioucas-Dias and M. A. T. Figueiredo. Multiplicative noise removal using variable splitting and constrained optimization. *IEEE Trans. Image Process.*, 19(7):1720–1730, 2010.

[20] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, stastical, and sparse regression-based approaches. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 5(2):354–379, Apr. 2012.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed optimization and statistical learning via the alternating direction method of multipliers*, volume 3. Found. Trends Mach. Learn., Jan. 2011.

[22] K. Bredies and M. Holler. Higher-order total variation approaches and generalisations. *Inverse Problems*, 36(12):123001, Dec. 2020.

[23] H. Bristow, A. Eriksson, and S. Lucey. Fast convolutional sparse coding. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 391–398, June 2013.

[24] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye. Data-driven tight frame construction and image denoising. *Appl. Comput. Harmon. Anal.*, 37(1):89–105, 2014.

[25] E. J. Candes, X. Li, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, Jun. 2011.

[26] X. Cao, L. Yang, and X. Guo. Total variation regularized rpca for irregularly moving object detection under dynamic background. *IEEE Trans. Cybern.*, 46(4):1014–1027, Apr. 2016.

[27] Y. Cao, 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.*, 26(12):2176–2188, Dec. 2016.

[28] H. Carfantan and J. Idier. Statistical linear destriping of satellite-based pushbroom-type images. *IEEE Trans. Geosci. Remote Sens.*, 48(4):1860–1871, Feb. 2010.

[29] M. Carlavan and L. Blanc-Féraud. Sparse Poisson noisy image deblurring. *IEEE Trans. Image Process.*, 21(4):1834–1846, Apr. 2012.

[30] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imag. Vis.*, 20:89–97, 2004.

[31] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock. An introduction to total variation for image analysis. In *Theoretical foundations and numerical methods for sparse recovery*, pages 263–340. de Gruyter, 2010.

[32] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vis.*, 40(1):120–145, 2011.

[33] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen. An augmented lagrangian method for total variation video restoration. *IEEE Trans. Image Process.*, 20(11):3097–3111, Nov. 2011.

[34] C-I Chang. *Hyperspectral imaging: techniques for spectral detection and classification*, volume 1. Springer Science & Business Media, 2003.

[35] X.-K. Chang, J. Yang, and H. Zhang. Golden ratio primal-dual algorithm with linesearch. *SIAM J. Optim.*, 32(3):1584–1613, 2022.

[36] Y. Chang, L. Yan, H. Fang, and C. Luo. Anisotropic spectral-spatial total variation model for multispectral remote sensing image destriping. *IEEE Trans. Image Process.*, 24(6):1852–1866, Jun. 2015.

[37] Y. Chang, L. Yan, H. Fang, S. Zhong, and W. Liao. Hsi-denet: Hyperspectral image restoration via convolutional neural network. *IEEE Trans. Geosci. Remote Sens.*, 57(2):667–682, Feb. 2019.

[38] Y. Chang, L. Yan, L. Liu, H. Fang, and S. Zhong. Infrared aerothermal nonuniform correction via deep multiscale residual network. *IEEE Geosci. Remote Sens. Lett.*, 16(7):1120–1124, Jul. 2019.

[39] Y. Chang, L. Yan, T. Wu, and S. Zhong. Remote sensing image stripe noise removal: From image decomposition perspective. *IEEE Trans. Geosci. Remote Sens.*, 54(12):7018–7031, Dec. 2016.

[40] R. Chartrand. Nonconvex splitting for regularized low-rank + sparse decomposition. *IEEE Trans. Signal Process.*, 60(11):5810–5819, Nov. 2012.

[41] Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, and G. He. Denoising of hyperspectral images using nonconvex low rank matrix approximation. *IEEE Trans. Geosci. Remote Sens.*, 55(9):5366–5380, Sep. 2017.

[42] 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. *Neurocomput.*, 267(6):95–106, Dec. 2017.

[43] Y. Chen, Huang T.-Z, X.-L. Zhao, L.-J. Deng, and J. Huang. Stripe noise removal of remote sensing images by total variation regularization and group sparsity constraint. *Remote Sens.*, 9(6):559, Jun. 2017.

[44] Y. Chen, Y. Zhou, W. Chen, S. Zu, W. Huang, and D. Zhang. Empirical low-rank approximation for seismic noise attenuation. *IEEE Trans. Geosci. Remote Sens.*, 55(8):4696–4711, 2017.

[45] G. Chierchia, E. Chouzenoux, P. L. Combettes, and J.-C. Pesquet. The proximity operator repository. *User's guide http://proximityoperator.net/download/guide.pdf (accessed October 3rd, 2021)*, 2020.

[46] G. Chierchia, D. Cozzolino, G. Poggi, and L. Verdoliva. SAR image despeckling through convolutional neural networks. In *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 5438–5441, 2017.

[47] G. Chierchia, N. Pustelnik, J.C. Pesquet, and B. Pesquet-Popescu. Epigraphical projection and proximal tools for solving constrained convex optimization problems. *Signal, Image Video Process.*, 9(8):1737–1749, 2015.

[48] P. L Combettes and J.-C. Pesquet. Fixed point strategies in data science. *IEEE Trans. Signal Process.*, 69:3878–3905, 2021.

[49] P. L. Combettes and N. N. Reyes. Moreau's decomposition in banach spaces. *Math. Program.*, 139:103–114, Jun. 2013.

[50] P. L. Combettes and B. C. Vũ. Variable metric quasi-fejér monotonicity. *Nonlin. Anal.*, 78:17–31, Feb. 2013.

[51] L. Condat. A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *J. Opt. Theory Appl.*, 158(2):460–479, 2013.

[52] L. Condat. A generic proximal algorithm for convex optimization-application to total variation minimization. *IEEE Signal Process. Lett.*, 21(8):985–989, Aug. 2014.

[53] L. Condat. Fast projection onto the simplex and the l1 ball. *Math. Program.*, 158(1):575–585, Jul. 2016.

[54] C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré. Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection. *SIAM J. Imag. Sci.*, 7(4):2448–2487, 2014.

[55] W. Dong, T. Huang, G. Shi, Y. Ma, and X. Li. Robust tensor approximation with laplacian scale mixture modeling for multiframe image and video denoising. *IEEE J. Sel. Topics Signal Process.*, 12(6):1435–1448, 2018.

[56] D.L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory*, 41(3):613–627, 1995.

[57] 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 Sensing*, 10(3):361, Feb. 2018.

[58] J. Eckstein and D. Bertsekas. On the douglas-rachford splitting method and proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(3), Apr. 1992.

[59] C. Elachi and J. J. Van Zyl. *Introduction to the physics and techniques of remote sensing*. John Wiley & Sons, 2021.

[60] H. Fan, Y. Chen, H. Zhang, and G. Kuang. Hyperspectral image restoration using low-rank tensor recovery. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 10(10):4589–4604, Oct. 2017.

[61] H. Fan, C. Li, Y. Guo, G. Kuang, and J. Ma. Spatial-spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.*, 56(10):6196–6213, Oct. 2018.

[62] Y. Fan, K. Wang, J. Li, G. Zhou, B. Zhang, and Y. Wu. L-hypersurface based parameters selection in composite regularization models with application to SAR and TomoSAR imaging. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 16:8297–8309, 2023.

[63] X.-R. Feng, H.-C. Li, J. Li, Q. Du, A. Plaza, and W. J. Emery. Hyperspectral unmixing using sparsity-constrained deep nonnegative matrix factorization with total variation. *IEEE Trans. Geosci. Remote Sens.*, 56(10):6245–6257, 2018.

[64] X.-R. Feng, H.-C. Li, R. Wang, Q. Du, X. Jia, and A. Plaza. Hyperspectral unmixing based on nonnegative matrix factorization: A comprehensive review. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 15:4414–4436, 2022.

[65] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.*, 2(1):17–40, 1976.

[66] L. Gao, B. Zhao, X. Jia, W. Liao, and B. Zhang. Optimized kernel minimum noise fraction transformation for hyperspectral image classification. *Remote Sensing*, 9(6), Apr. 2017.

[67] C. Garcia-Cardona and B. Wohlberg. Convolutional dictionary learning: A comparative review and new algorithms. *IEEE Trans. Comput. Imag.*, 4(3):366–381, Sep. 2018.

[68] P. Ghamisi, N. Yokoya, J. Li, W. Liao, S. Liu, J. Plaza, B. Rasti, and A. Plaza. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geosci. Remote Sens. Mag.*, 5(4):37–78, 2017.

[69] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, 7(3):1005–1028, 2009.

[70] T. Goldstein, M. Li, and X. Yuan. Adaptive primal-dual splitting methods for statistical learning and image processing. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Proc. Advances Neural Inf. Process. Syst.*, volume 28, pages 2089–2097. Curran Associates, Inc., 2015.

[71] M. Halicek, H. Fabelo, S. Ortega, G. M. Callico, and B. Fei. In-vivo and ex-vivo tissue analysis through hyperspectral imaging techniques: Revealing the invisible features of cancer. *Cancers*, 11(6), 2019.

[72] Per Christian Hansen. Analysis of discrete ill-posed problems by means of the L-curve. *SIAM rev.*, 34(4):561–580, 1992.

[73] B. He, F. Ma, S. Xu, and X. Yuan. A generalized primal-dual algorithm with improved convergence condition for saddle point problems. *SIAM J. Imag. Sci.*, 15(3):1157–1183, 2022.

[74] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2016.

[75] 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.*, 11(3):713–729, Mar. 2018.

[76] W He, H Zhang, L Zhang, and H Shen. Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration. *IEEE Trans. Geosci. Remote Sens.*, 54(1):178–188, 2015.

[77] D. Hong, L. Gao, J. Yao, N. Yokoya, J. Chanussot, U. Heiden, and B. Zhang. Endmember-guided unmixing network (EGU-Net): A general deep learning framework for self-supervised hyperspectral unmixing. *IEEE Trans. Neural Netw. Learn. Syst.*, 33(11):6518–6531, Nov. 2022.

[78] 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.*, 59(7):5966–5978, Jul. 2021.

[79] M. S. Hosseini and K. N. Plataniotis. High-accuracy total variation with application to compressed video sensing. *IEEE Trans. Image Process.*, 23(9):3869–3884, 2014.

[80] T. Hu, W. Li, N. Liu, R. Tao, F. Zhang, and P. Scheunders. Hyperspectral image restoration using adaptive anisotropy total variation and nuclear norms. *IEEE Trans. Geosci. Remote Sens.*, 59(2):1516–1533, Feb. 2021.

[81] J. Huang, W.-C. Di, J.-J. Wang, J. Lin, and T.-Z. Huang. Bilateral joint-sparse regression for hyperspectral unmixing. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 14:10147–10161, Sep. 2021.

[82] J. Huang, T. Huang, L. Deng, and X. Zhao. Joint-sparse-blocks and low-rank representation for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 57(4):2419–2438, 2018.

[83] Y. Huang, G. Liao, J. Xu, J. Li, and D. Yang. GMTI and parameter estimation for MIMO SAR system via fast interferometry RPCA method. *IEEE Trans. Geosci. Remote Sens.*, 56(3):1774–1787, 2018.

[84] Y. Huang, L. Zhang, J. Li, Z. Chen, and X. Yang. Reweighted tensor factorization method for SAR narrowband and wideband interference mitigation using smoothing multiview tensor model. *IEEE Trans. Geosci. Remote Sens.*, 58(5):3298–3313, 2020.

[85] T. Ince. Hyperspectral image denoising using group low-rank and spatial-spectral total variation. *IEEE Access*, 7:52095–52109, Apr. 2019.

[86] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Sparse unmixing of hyperspectral data. *IEEE Trans. Geosci. Remote Sens.*, 49(6):2014–2039, 2011.

[87] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Collaborative sparse regression for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 52(1):341–354, Jan. 2014.

[88] M.-D. Iordache, J.M. Bioucas-Dias, and A. Plaza. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 50(11):4484–4502, 2012.

[89] R. Isono, K. Naganuma, and S. Ono. Robust spatiotemporal fusion of satellite images: A constrained convex optimization approach. *arXiv:2308.00500*, 2023.

[90] R. Isono, K. Naganuma, and S. Ono. Robust spatiotemporal fusion of satellite images via convex optimization. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 1–5, 2023.

[91] H. Ji, S. Huang, Z. Shen, and Y. Xu. Robust video restoration by joint sparse and low rank matrix approximation. *SIAM J. Imag. Sci.*, 4(4):1122–1142, 2011.

[92] T.-X. Jiang, L. Zhuang, T.-Z. Huang, X.-L. Zhao, and J. M. Bioucas-Dias. Adaptive hyperspectral mixed noise removal. *IEEE Trans. Geosci. Remote Sens.*, early access, vol. 60, Jun. 2021, Art no. 5511413, doi: 10.1109/TGRS.2021.3085779.

[93] K. Naganuma K. Sato and S. Ono. Enhancing hyperspectral anomaly detection by difference-of-convex sparse anomaly modeling. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Arp. 2024. (Accepted).

[94] A. Katsuma, S. Kyochi, S. Ono, and I. Selesnick. Epigraphically-relaxed linearly-involved generalized Moreau-enhanced model for layered mixed norm regularization. In *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pages 2240–2244, 2023.

[95] N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Process. Mag.*, 19(1):44–57, 2002.

[96] T. H. Kim and R. M. Narayanan. Compressive sensing based image reconstruction for synthetic aperture radar using discrete cosine transform and noiselets. In *Proc. Int. Conf. Telecommun. Signal Process.*, pages 582–586, 2015.

[97] N. Komodakis and J.-C. Pesquet. Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Process. Mag.*, 32(6):31–54, Nov. 2015.

[98] J. Kovacevic and A. Chebira. Life beyond bases: The advent of frames (part I). *IEEE Signal Process. Mag.*, 24(4):86–104, Jul. 2007.

[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Adv. Neural Inf. Process. Syst.*, volume 25, page 1097–1105. Curran Associates, Inc., 2012.

[100] X. Kuang, X. Sui, Y. Liu, Q. Chen, and G. Gu. Single infrared image optical noise removal using deep convolutional networks. *IEEE Photon. J.*, 10(2):1–15, Apr. 2018.

[101] S. Kyochi, S. Ono, and I. Selesnick. Epigraphical relaxation for minimizing layered mixed norms. *IEEE Trans. Signal Process.*, 69:2923–2938, 2021.

[102] D. Landgrebe. Hyperspectral image data analysis. *IEEE Signal Process. Mag.*, 19(1):17–28, 2002.

[103] S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017.

[104] B. Li, Y.-K. Lai, and P. L. Rosin. Sparse graph regularized mesh color edit propagation. *IEEE Trans. Image Process.*, 29:5408–5419, 2020.

[105] H Li, R Feng, L Wang, Y Zhong, and L Zhang. Superpixel-based reweighted low-rank and total variation sparse unmixing for hyperspectral remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.*, 59(1):629–647, 2020.

[106] H.-C. Li, X.-R. Feng, D.-H. Zhai, Q. Du, and A. Plaza. Self-supervised robust deep matrix factorization for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 60:1–14, 2022. Art no. 5513214.

[107] L. Li, W. Li, Q. Du, and R. Tao. Low-rank and sparse decomposition with mixture of gaussian for hyperspectral anomaly detection. *IEEE Trans. Cybern.*, 51(9):4363–4372, 2021.

[108] X. Li, V. Monga, and A. Mahalanobis. Multiview automatic target recognition for infrared imagery using collaborative sparse priors. *IEEE Trans. Geosci. Remote Sens.*, 58(10):6776–6790, Oct. 2020.

[109] Z. Li, F. Malgouyres, and T. Zeng. Regularized non-local total variation and application in image restoration. *J. Math. Imag. Vis.*, 59(2):296–317, 2017.

[110] Y. Liang, H. Zheng, G. Yang, Q. Du, and H. Su. Superpixel-based weighted sparse regression and spectral similarity constrained for hyperspectral unmixing. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 16:6825–6842, Jul. 2023.

[111] G. Licciardi, F. Pacifici, D. Tuia, Prasad, T. West, F. Giacco, and et al. Decision fusion for the classification of hyperspectral data: Outcome of the 2008 grs-s data fusion contest. *IEEE Trans. Geosci. Remote Sens.*, 47(11):3857–3865, 2009.

[112] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):171–184, 2013.

[113] 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.*, 58(3):1777–1789, Mar. 2020.

[114] 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.*, 57(12):10310–10321, Dec. 2019.

[115] P. Liu, M. Wang, L. Wang, and W. Han. Remote-sensing image denoising with multi-sourced information. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 12(2):660–674, Feb. 2019.

[116] 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.*, 54(5):3049–3060, May 2016.

[117] 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.*, 56(2):808–822, Feb. 2018.

[118] Y. Liu, Y. Xu, and W. Yin. Acceleration of primal-dual methods by preconditioning and simple subproblem procedures. *J. Sci. Comput.*, 86(21):1–34, Jan. 2021.

[119] J. Lou and Y.-M. Cheung. Robust low-rank tensor minimization via a new tensor spectral k -support norm. *IEEE Trans. Image Process.*, 29:2314–2327, 2020.

[120] B. Lu, P. D. Dao, J. Liu, Y. He, and J. Shang. Recent advances of hyperspectral imaging technology and applications in agriculture. *Remote Sens.*, 12(16):2659, 2020.

[121] 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 *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2724–2727, Jun. 2014.

[122] Z. Lu, Q. Ling, H. Li, and W. Li. Video restoration based on a novel second order nonlocal total variation model. *Signal Process.*, 133:79–96, 2017.

[123] W. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C. Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Process. Mag.*, 31(1):67–81, Jan. 2014.

[124] W.K. Ma, J. M. Bioucas-Dias, T.H. Chan, N. Gillis, P. Gader, A. Plaza, A. Ambikapathi, and C.Y. Chi. A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Process. Mag.*, 31(1):67–81, Jan. 2014.

[125] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geosci. Remote Sens.*, 55(2):645–657, Feb. 2017.

[126] N. K. Malakar, G. C. Hulley, S. J. Hook, K. Laraby, M. Cook, and J. R. Schott. An operational land surface temperature product for landsat thermal data: Methodology and validation. *IEEE Trans. Geosci. Remote Sens.*, 56(10):5717–5735, 2018.

[127] Y. Malitsky and T. Pock. A first-order primal-dual algorithm with linesearch. *SIAM J. Optim.*, 28(1):411–432, 2018.

[128] B. E. Moore, C. Gao, and R. R. Nadakuditi. Panoramic robust pca for foreground-background separation on noisy, free-motion camera video. *IEEE Trans. Comput. Imag.*, 5(2):195–211, Jun. 2019.

[129] M. Moradikia, S. Samadi, and M. Cetin. Joint SAR imaging and multi-feature decomposition from 2-D under-sampled data via low-rankness plus sparsity priors. *IEEE Trans. Comput. Imag.*, 5(1):1–16, 2019.

[130] J. L. Mueller and S. Siltanen. *Linear and nonlinear inverse problems with practical applications*, volume 10. SIAM, 2012.

[131] B. Munch, P. Trtik, F. Marone, and M. Stampanoni. Stripe and ring artifact removal with combined wavelet-fourier filtering. *Opt. Express*, 17(10):8567–8591, May 2009.

[132] K. Naganuma and S. Ono. A general destriping framework for remote sensing images using flatness constraint. *IEEE Trans. Geosci. and Remote Sens.*, 60:1–16, Feb. 2022. Art no. 5525016.

[133] K. Naganuma and S. Ono. Cross-frame foreground structural similarity modeling by convolutional sparse representation. In *Proc. Asia-Pac. Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, pages 779–783, 2023.

[134] K. Naganuma and S. Ono. Static-scene constrained optimization for matrix/tensor-decomposition-free foreground-background separation. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 1–5, 2023.

[135] Kazuki Naganuma and Shunsuke Ono. Variable-wise diagonal preconditioning for primal-dual splitting: Design and applications. *IEEE Trans. Signal Process.*, 71:3281–3295, 2023.

[136] S. Ono. l_0 gradient projection. *IEEE Trans. Image Process.*, 26(4):1554–1564, 2017.

[137] S. Ono. Primal-dual plug-and-play image restoration. *IEEE Signal Process. Lett.*, 24(8):1108–1112, Aug. 2017.

[138] S. Ono. Efficient constrained signal reconstruction by randomized epigraphical projection. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 4993–4997. IEEE, 2019.

[139] S. Ono and I. Yamada. Second-order total generalized variation constraint. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 4938–4942, 2014.

[140] S. Ono and I. Yamada. Hierarchical convex optimization with primal-dual splitting. *IEEE Trans. Signal Process.*, 63(2):373–388, Jan. 2015.

[141] S. Ono and I. Yamada. Signal recovery with certain involved convex data-fidelity constraints. *IEEE Trans. Signal Process.*, 63(22):6149–6163, Nov. 2015.

[142] S. Ono, I. Yamada, and I. Kumazawa. Total generalized variation for graph signals. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 5456–5460, 2015.

[143] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proc. the IEEE*, 106(5):808–828, 2018.

[144] S. Ozkan, B. Kaya, and B. Akar, G. EndNet: Sparse autoencoder network for endmember extraction and hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 57(1):482–496, 2018.

[145] A. Parekh and I. W. Selesnick. Convex denoising using non-convex tight frame regularization. *IEEE Signal Process. Lett.*, 22(10):1786–1790, Oct. 2015.

[146] N. Parikh and S. Boyd. Proximal algorithms. *Found. Trends Mach Learn.*, 1(3):127–239, 2014.

[147] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond. GSPBOX: A toolbox for signal processing on graphs. *ArXiv e-prints*, 2014.

[148] T. Pock and A. Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1762–1769, Nov. 2011.

[149] B. Rasti and B. Koirala. SUnCNN: Sparse unmixing using unsupervised convolutional neural network. *IEEE Geosci. Remote Sens. Lett.*, 19:1–5, 2022. Art no. 5508205.

[150] B. Rasti, B. Koirala, P. Scheunders, and J. Chanussot. MiSiCNet: Minimum simplex convolutional network for deep hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 60:1–15, Jan. 2022.

[151] B. Rasti, B. Koirala, P. Scheunders, and P. Ghamisi. UnDIP: Hyperspectral unmixing using deep image prior. *IEEE Trans. Geosci. Remote Sens.*, 60:1–15, 2022. Art no. 5504615.

[152] B. Rasti, P. Scheunders, P. Ghamisi, G. Licciardi, and J. Chanussot. Noise reduction in hyperspectral imagery: Overview and application. *Remote Sensing*, 10(3):482, 2018.

[153] B. Rasti, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson. Wavelet based hyperspectral image restoration using spatial and spectral penalties. In *Proc. Image Signal Process. for Remote Sens.*, volume 8892, pages 135–142. SPIE, 2013.

[154] Behnood Rasti, Yi Chang, Emanuele Dalsasso, Loic Denis, and Pedram Ghamisi. Image restoration for remote sensing: Overview and toolbox. *IEEE Geosci. Remote Sens. Mag.*, 10(2):201–230, 2022.

[155] L. Ren, Z. Ma, F. Bovolo, and L. Bruzzone. A nonconvex framework for sparse unmixing incorporating the group structure of the spectral library. *IEEE Trans. Geosci. Remote Sens.*, 60:1–19, 2021.

[156] O. Riou, S. Berrebi, and P. Bremond. Nonuniformity correction and thermal drift compensation of thermal infrared camera. In *Proc. SPOE, Thermosense XXVI*, volume 5405, pages 294–302, Apr. 2004.

[157] C. Rodarmel and J. Shan. Principal component analysis for hyperspectral image classification. *Surv. Land Inf. Sci.*, 62(2):115–122, 2002.

[158] A. Romero, C. Gatta, and G. Camps-Valls. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.*, 54(3):1349–1362, Mar. 2016.

[159] A. Sandryhaila and J. M. F. Moura. Discrete signal processing on graphs. *IEEE Trans. Signal Process.*, 61(7):1644–1656, 2013.

[160] K. Sato and S. Ono. Robust hyperspectral anomaly detection with simultaneous mixed noise removal via constrained convex optimization. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 1–5, 2023.

[161] G. Shaw and D. Manolakis. Signal processing for hyperspectral image exploitation. *IEEE Signal Process. Mag.*, 19(1):12–16, 2002.

[162] H. Shen, X. Li, L. Zhang, D. Tao, and C. Zeng. Compressed sensing-based inpainting of aqua moderate resolution imaging spectroradiometer band 6 using adaptive spectrum-weighted sparse bayesian dictionary learning. *IEEE Trans. Geosci. Remote Sens.*, 52(2):894–906, Feb. 2014.

[163] X Shen, H Liu, X Zhang, K Qin, and X Zhou. Superpixel-guided local sparsity prior for hyperspectral sparse regression unmixing. *IEEE Geosci. Remote Sens. Lett.*, 19:1–5, 2022. Art no. 6015105.

[164] 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. *Infr. Phys. Technol.*, 76:691–697, May 2016.

[165] D. I Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process. Mag.*, 30(3):83–98, May 2013.

[166] O. Sidorov and J. Y. Hardeberg. Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 3844–3851, Oct 2019.

[167] J. Song, J.-H. Jeong, D.-S. Park, H.-H. Kim, D.-C. Seo, and J.-C. Ye. Unsupervised denoising for satellite imagery using wavelet directional cyclegan. *IEEE Trans. Geosci. Remote Sens.*, 59(8):6823–6839, Aug. 2021.

[168] J. C. O Souza, P. R. Oliveira, and A. Soubeyran. Global convergence of a proximal linearized algorithm for difference of convex functions. *Optim. Lett.*, 10(7):1529–1539, 2016.

[169] C. M. Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, 9(6):1135–1151, 1981.

[170] H. Su, Z. Wu, H. Zhang, and Q. Du. Hyperspectral anomaly detection: A survey. *IEEE Geosci. Remote Sens. Mag.*, 10(1):64–90, 2022.

[171] D.-W. Sun. *Hyperspectral imaging for food quality analysis and control.* Elsevier, 2010.

[172] Y. Sun, L. Lei, D. Guan, X. Li, and G. Kuang. SAR image speckle reduction based on nonconvex hybrid total variation model. *IEEE Trans. Geosci. Remote Sens.*, 59(2):1231–1249, 2021.

[173] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1–9, 2015.

[174] S. Takemoto, K. Naganuma, and S. Ono. Graph spatio-spectral total variation model for hyperspectral image denoising. *IEEE Geosci. Remote Sens. Lett.*, 19:1–5, Jul. 2022. Art no. 6012405.

[175] S. Takemoto and S. Ono. Enhancing spatio-spectral regularization by structure tensor modeling for hyperspectral image denoising. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 1–5, 2023.

[176] S. Takeyama and S. Ono. Robust hyperspectral image fusion with simultaneous guide image denoising via constrained convex optimization. *IEEE Trans. Geosci. Remote Sens.*, 60:1–18, 2022.

[177] S Takeyama, S Ono, and I Kumazawa. A constrained convex optimization approach to hyperspectral image restoration with hybrid spatio-spectral regularization. *Remote Sens.*, 12(21):3541, 2020.

[178] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung. Sampling signals on graphs: From theory to applications. *IEEE Signal Process. Mag.*, 37(6):14–30, 2020.

[179] P. S Thenkabail and J. G Lyon. *Hyperspectral remote sensing of vegetation*. CRC press, 2016.

[180] A. J. Tom and S. N. George. Video completion and simultaneous moving object detection for extreme surveillance environments. *IEEE Signal Process. Lett.*, 26(4):577–581, Apr. 2019.

[181] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018.

[182] B. C. Vu. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.*, 38(3):667–681, 2013.

[183] J.-J. Wang, T.-Z. Huang, J. Huang, H.-X. Dou, L.-J. Deng, and X.-L. Zhao. Row-sparsity spectral unmixing via total variation. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 12(12):5009–5022, Dec. 2019.

[184] M. Wang, Q. Wang, J. Chanussot, and D. Hong. l_0-l_1 hybrid total variation regularization and its applications on hyperspectral image mixed noise removal and compressed sensing. *IEEE Trans. Geosci. Remote Sens.*, 59(9):7695–7710, Sep. 2021.

[185] M. Wang, Q. Wang, J. Chanussot, and D. Li. Hyperspectral image mixed noise removal based on multidirectional low-rank modeling and spatial–spectral total variation. *IEEE Trans. Geosci. Remote Sens.*, 59(1):488–507, 2021.

[186] R Wang, H.C. Li, A Pizurica, J Li, A Plaza, and W.J. Emery. Hyperspectral unmixing using double reweighted sparse regression and total variation. *IEEE Geosci. Remote Sens. Lett.*, 14(7):1146–1150, 2017.

[187] Y. Wang, Y. Y. Tang, C. Zou, and L. Yang. Spectral-spatial hyperspectral image destriping using sparse learning and spatial unidirection prior. In *Proc. IEEE Int. Conf. Cybern. (CYBCONF)*, pages 1–5, Jul. 2017.

[188] 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.*, 13(4):600–612, Apr. 2004.

[189] M. Wen, J. Peng, C. Zhu, S. Yue, and Y. Tang. A preconditioning technique for first-order primal-dual splitting method in convex optimization. *Math. Problems Eng.*, 2017, 2017.

[190] Qihao Weng. Thermal infrared remote sensing for urban climate and environmental studies: Methods, applications, and trends. *ISPRS J. Photogram. Remote Sens.*, 64(4):335–344, 2009.

[191] B. Wohlberg. Efficient convolutional sparse coding. In *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 7173–7177, 2014.

[192] B. Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Trans. Image Process.*, 25(1):301–315, Jan. 2016.

[193] D. Wu and D.-W. Sun. Advanced applications of hyperspectral imaging technology for food quality and safety analysis and assessment: A review — part i: Fundamentals. *Innov. Food Sci. Emerg. Technol.*, 19:1–14, 2013.

[194] L. Wu, J. Huang, and M.-S. Guo. Multidimensional low-rank representation for sparse hyperspectral unmixing. *IEEE Geosci. Remote Sens. Lett.*, 20:1–5, Mar. 2023. Art no. 5502805.

[195] 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. (CVPR)*. IEEE, Jun.

[196] 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)*, pages 1–8, Dec. 2009.

[197] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1948–1955, Jun. 2012.

[198] Y. Xu and W. Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM J. Imag. Sci.*, 6(3):1758–1789, 2013.

[199] J. Xue, Y. Zhao, W. Liao, and J. C. Chan. Nonlocal low-rank regularized tensor decomposition for hyperspectral image denoising. *IEEE Trans. Geosci. Remote Sens.*, 57(7):5174–5189, Jul. 2019.

[200] S. Yang, M. Wang, P. Li, L. Jin, B. Wu, and L. Jiao. Compressive hyperspectral imaging via sparse tensor and nonlinear compressed sensing. *IEEE Trans. Geosci. Remote Sens.*, 53(11):5943–5957, 2015.

[201] H. Ye, H. Li, B. Yang, F. Cao, and Y. Tang. A novel rank approximation method for mixture noise removal of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.*, 57(7):4457–4469, 2019.

[202] Z. Ye, T. Möllenhoff, T. Wu, and D. Cremers. Optimization of graph total variation via active-set-based combinatorial reconditioning. In Silvia Chiappa and Roberto Calandra, editors, *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, volume 108 of *Proceedings of Machine Learning Research*, pages 657–668. PMLR, 26–28 Aug. 2020.

[203] N. Yokoya, C. Grohnfeldt, and J. Chanussot. Hyperspectral and multispectral data fusion: A comparative review of the recent literature. *IEEE Trans. Geosci. Remote Sens. Mag.*, 5(2):29–56, 2017.

[204] Q. Yuan, L. Zhang, and H. Shen. Hyperspectral image denoising employing a spectral-spatial adaptive total variation model. *IEEE Trans. Geosci. Remote Sens.*, 50(10):3660–3677, Oct. 2012.

[205] Y. Yuan, Z. Zhang, and Q. Wang. Improved collaborative non-negative matrix factorization and total variation for hyperspectral unmixing. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 13:998–1010, Mar. 2020.

[206] R. Yuzuriha, R. Kurihara, M. Okuda, and R. Matsuoka. Total nuclear norms of gradients for hyperspectral image pansharpening. In *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pages 2695–2698, 2020.

[207] 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.*, 17(6):1072–1076, Jun. 2020.

[208] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Trans. Geosci. Remote Sens.*, 52(8):4729–4743, Aug. 2014.

[209] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, 2017.

[210] L. Zhang, Y. Qian, J. Han, P. Duan, and P. Ghamisi. Mixed noise removal for hyperspectral image with l_0-l_{1-2}sstv regularization. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, 15:5371–5387, Jun. 2022.

[211] Q. Zhang, Q. Yuan, J. Li, Z. Yang, and X. Ma. Learning a dilated residual network for SAR image despeckling. *Remote Sens.*, 10(2), 2018.

[212] S Zhang, J Li, C Li, H, C Deng, and A Plaza. Spectral–spatial weighted sparse regression for hyperspectral image unmixing. *IEEE Trans. Geosci. Remote Sens.*, 56(6):3265–3276, 2018.

[213] X. Zhang, Y. Sun, J. Zhang, P. Wu, and L. Jiao. Hyperspectral unmixing via deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.*, 15(11):1755–1759, 2018.

[214] M. Zhao, L. Cheng, X. Yang, P. Feng, L. Liu, and N. Wu. TBC-Net: A real-time detector for infrared small target detection using semantic constraint, 2019.

[215] M. Zhao, X. Wang, J. Chen, and W. Chen. A plug-and-play priors framework for hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.*, 60:1–13, 2022. Art no. 5501213.

[216] Y. Zhao and J. Yang. Hyperspectral image denoising via sparse representation and low-rank constraint. *IEEE Trans. Geosci. Remote Sens.*, 53(1):296–308, 2014.

[217] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, Y. Chen, and W. He. Double-factor-regularized low-rank tensor factorization for mixed noise removal in hyperspectral image. *IEEE Trans. Geosci. Remote Sens.*, 58(12):8450–8464, Dec. 2020.

[218] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, and T.-Y. Ji. Mixed noise removal in hyperspectral image via low-fibered-rank regularization. *IEEE Trans. Geosci. Remote Sens.*, 58(1):734–749, Jan. 2020.

[219] W. Zhu, V. Chayes, A. Tiard, S. Sanchez, D. Dahlberg, A. L. Bertozzi, S. Osher, D. Zosso, and D. Kuang. Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm. *IEEE Trans. Geosci. Remote Sens.*, 55(5):2786–2798, 2017.

Publications Related to This Dissertation

Articles in Journal Papers

[J1] K. Naganuma and S. Ono, "A general destriping framework for remote sensing images using flatness constraint," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-16, 2022, Art no. 5525016.

[J2] K. Naganuma and S. Ono, 'Variable-wise diagonal preconditioning for primal-dual splitting: Design and applications," *IEEE Transactions on Signal Processing*, vol. 71, pp. 3281-3295, 2023.

[J3] K. Naganuma and S. Ono, "Towards Robust Hyperspectral Unmixing: Mixed Noise Modeling and Image-Domain Regularization," submitted to IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. (under "Major Revision" in Dec. 2023)

International Conference

[C1] K. Naganuma, S. Takeyama, and S. Ono, "Zero-gradient constraints for destriping of remote-sensing data," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toronto, Canada, pp. 1480-1484, Jun. 2021.

[C2] K. Naganuma and S. Ono, "Operator-norm-based variable-wise diagonal preconditioning for automatic stepsize selection of a primal-dual splitting algorithm," in *Proceedings of European Signal Processing Conference (EUSIPCO)*, Belgrade, Servia, pp. 1480-1484, Aug. 2022.

Other Publications

Articles in Journal Papers

[J4] K. Naganuma, T. Suzuki, H. Tsuji, and T. Kimura, "Study of safe elliptic curve cryptography over Gaussian integer," *IEICE Transactions on Fundamentals*, vol. E103-A, no. 12, pp. 1624-1628, Dec. 2020.

[J5] T. Suzuki, K. Naganuma, H. Tsuji, and T. Kimura, "Low-level standard deviation estimation for Gaussian noised images by denoising filter and Gaussian noise re-superimposition," *IEICE Transactions on Fundamentals*, vol. J104-A, no. 7, pp. 160-172, Jul. 2021. (in Japanese)

[J6] S. Takemoto, K. Naganuma, and S. Ono, "Graph spatio-spectral total variation for hyperspectral image denoising," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1-5, Jul. 2022, Art no. 6012405.

[J7] S. Ono, K. Naganuma, and K. Yamashita, "Graph signal sampling under smoothness priors: A difference-of-convex approach," submitted to IEEE Signal Processing Letter.

[J8] R. Isono, K. Naganuma, and S. Ono, "Robust spatiotemporal fusion of satellite images: A constrained convex optimization approach," submitted to IEEE Transactions on Geoscience and Remote Sensing. (under "Major Revision" in Dec. 2023)

International Conference

[C3] K. Naganuma, T. Suzuki, H. Tsuji, and T. Kimura, "Implementation of the elliptic curve cryptography over Gaussian integral finite group onto excel," in *Proceedings of IEICE International Workshop on Smart Info-Media System in Asia (SISA)*, Tokyo, Japan, vol. 57, pp. 119-124, Sep. 2019.

[C4] K. Naganuma and S. Ono, "Static-scene constrained optimization for matrix/tensor-decomposition-free foreground-background separation," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, pp. 1-5, Jun. 2023.

[C5] R. Isono, K. Naganuma, and S. Ono, "Robust spatio-temporal fusion of satellite images via convex optimization," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Rhodes Island, Greece, pp. 1-5, Jun. 2023.

[C6] K. Naganuma and S. Ono, "Cross-frame foreground structural similarity modeling by convolutional sparse representation," (invited paper) in *Proceedings of APSIPA Annual Summit and Conference (APSIPA ASC)*, Taipei, Taiwan, pp. 779-783, Oct. 2023.

[C7] K. Sato, K. Naganuma, and S. Ono, "Enhancing hyperspectral anomaly detection by difference-of-convex sparse anomaly modeling," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seoul, Korea, Arp. 2024. (Accepted)