# Compressive sensing sample statistics estimation from low-dimensional projections

M.Sc. Jonathan Arley Monsalve Salazar

Doctoral thesis to qualify for the Doctor of Engineering degree, electronic area

Advisor

Henry Arguello

Ph.D. in Electrical and Computer Engineering

Universidad Industrial de Santander

School of Physical-Mechanical Engineering

Department of Electrical, Electronics, and Telecommunication Engineering.

Bucaramanga

2023

## Dedicatory

I dedicate my doctoral thesis to my parents, family, and friends, who have brought me unconditional support over all the years.

**Content**

## List of Figures

**page**

## List of Tables

**page**

## List of Algorithms

## List of Appendices

## List of Abbreviations

**SI** Spectral Imaging

**HSI** Hyperspectral Imaging

**CS** Compressive Sensing

**CSI** Compressive Spectral Imaging

**CCS** Compressive Covariance Sampling

**PCA** Principal Component Analysis

**SVD** Singular Value Decomposition

**RIP** Restricted Isometry Property

**RP** Random Projection

**3D-CASSI** Spatial-spectral Coded Compressive Spectral Imager

**CM** Covariance Matrix

**CPPCA** Compressive-Projections Principal Component Analysis

**CoCoSVi** Compressive Covariance Spectral Video.

## Resumen

**Título:** Estimación de estadísticos muestrales desde proyecciones aleatorioas de baja dimensión [*]

**Autor:** Jonathan Arley Monsalve Salazar [**]

**Palabras Clave:** Adquisición compresiva de imágenes, Adquisición Compresiva de la matrix de Covarianza, Adquisición Compresiva de video spectral.

**Descripción:** El muestreo compresivo de la covarianza (MCC) tiene como objetivo recuperar el segundo momento estadístico de una señal a partir de un conjunto de proyecciones aleatorias de baja dimensión. En particular, CCS recupera la matriz de covarianza (MC) en lugar de la señal de alta dimensión, lo que representa una reducción significativa de los datos reconstruidos en aplicaciones tales como imágenes hiperespectrales, donde la MC suele ser algunos órdenes de magnitud más pequeño que la imagen. Además, la MC proporciona información sobre el subespacio de los datos útil para diseñar protocolos de detección, entrenar modelos para la clasificación o incluso reconstruir la señal. Esta tesis estudia la estimación y el uso del segundo momento estadístico de las imágenes hiperespectrales en la acquisición compresiva de imágenes espectrales (CSI). Por lo tanto, esta tesis propone un algoritmo para reconstruir el segundo momento estadístico a partir de proyecciones aleatorias de baja dimensión de imágenes hiperespectrales y un algoritmo para diseñar el protocolo de adquisición utilizando la MC. Para ello se propone un problema de optimización convexa, un algoritmo y una arquitectura óptica. Además, esta tesis presenta el análisis de las garantías de convergencia y algunas propiedades teóricas para asegurar una correcta reconstrucción. El algoritmo propuesto se prueba en tareas de clasificación y reconstrucción de imágenes hiperespectrales, incluida la estimación de la cobertura terrestre utilizando la MC recuperada.

---

[*] Tesis de doctorado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Eléctrica, Electrónica y de Telecomunicaciones. Doctorado en Ingeniería, Área Electrónica. Director: Henry Arguello Fuentes

## Abstract

**Title:** Compressive sensing sample statistics estimation from low-dimensional projections [*]

**Author:** Jonathan Arley Monsalve Salazar [**]

**Keywords:** Compressive Spectral Imaging, Compressive Covariance Sampling, Compressive Spectral video.

**Description:** Compressive Covariance Sampling (CCS) aims to recover the second-moment statistics of a signal from a set of low-dimensional random projections. In particular, CCS recovers the covariance matrix (CM) instead of the underlying high-dimensional signal, representing a significant reduction in the reconstructed data for applications such as hyperspectral imaging, where the CM is usually some orders of magnitude smaller than the image. Additionally, the CM provides information about the data's subspace which is helpful for designing sensing protocols, training models for classification, or even reconstructing the signal. This thesis studies estimating and using the second statistical moment of the hyperspectral images in compressive spectral imaging (CSI). Hence, this thesis proposes an algorithm for reconstructing the second statistic moment from low-dimensional random projections of hyperspectral images and an algorithm for designing the sensing protocol using the CM. A convex optimization problem, an algorithm, and an optical architecture that take advantage of this approach are proposed. Furthermore, this thesis presents the convergence guarantees analysis and some theoretical properties to ensure a correct reconstruction. The proposed algorithm is tested over hyperspectral image reconstruction and classification tasks, including land cover estimation using the recovered CM.

---

[*] Doctoral Thesis

[**] School of Physical-Mechanical Engineering. Department of Electrical, Electronics, and Telecommunication Engineering. Doctoral degree in Engineering, electronic area. Advisor: Henry Arguello Fuentes

## Introduction

Remote sensing devices and applications usually rely on images acquired from satellite or airborne cameras to collect data from difficult access places. Traditionally, these cameras capture high spatial resolution RGB data from the scene, allowing the identification of targets based on shape and color. However, the resolution is affected by sensor distance to the target and light propagation medium (such as the atmosphere), making the classification based on shape and color challenging (Manolakis et al., 2003). Hyperspectral imaging (HSI) is a tool for classification and target detection that is not based on shape but on spectrum analysis only (i.e., color analysis). In particular, HSI acquires a spectral signature of each spatial position instead of only three colors, as in RGB images. The spectral signature comprises hundreds of narrow bands of color, referred to as wavelengths, associated with the scene composition. Hence, a target can be identified by analyzing the shape of the spectral signature instead of the object's shape, which allows target identification even in a subpixel size (Manolakis et al., 2003, 2009).

HS image reconstruction is usually expensive because of the signal's dimension. Some applications do not even use the HS image but require estimated statistical parameters. For instance, tasks such as linear discriminant analysis or principal component analysis use the covariance matrix (CM) to learn a linear projector that reduces the dimensionality of the data while maximizing an $\ell_2$ based metric (Balakrishnama and Ganapathiraju, 1998; Fowler, 2009; Xanthopoulos et al., 2013). For instance, in target detection algorithms, the CM of the background is used to detect unusual spectral signatures, which is the objective (Manolakis et al., 2009). Adaptative detection

problems also use the CM to guide the sensing strategy to detect a signal embedded in perturbations (Besson et al., 2008b; Monsalve et al., 2020).

Therefore, estimating the Covariance Matrix of a HS image is a central problem in machine learning and signal processing applications. Traditionally, the CM is approximated using the sample covariance matrix estimator (Chen et al., 2014; Fowler, 2009), which requires many realizations to be accurate. In HSI applications, it implies that many pixels must be acquired before the CM estimation occurs. However, acquiring a large number of pixels is time-consuming due to the scanning approach used in image sensing. Additionally, the resulting image needs to be stored, transmitted, or processed, which is also complex due to its huge size. If not enough samples are available, another common approach to estimate the covariance matrix consists in shrinking the CM towards a simple estimator, which presents a lower variance resulting in a better estimation (Chen et al., 2010; Steland, 2018). However, if the simpler estimator has a large bias, shrinking the solution toward this estimator can lead to poor results.

To reduce the amount of data captured by hyperspectral cameras, compressive sensing captures a set of random measurements known as compressive measurements. This is done by applying a random pattern on the incident ligh field and capturing a two-dimensional projection. Although it reduces the amount of data, it also add complexity to extract information of this set of compressive measurements. A common approach consists in reconstruct the signal and then extract any usefull inforrmation such as statistical information or classification maps. However, another approach is to recover the statistical information from the measurements. Reconstructing the CM from compressive measurements also allows for recovering a low-dimensional approxi-

mation of the HS image based on the PCA strategy. This low-dimensional HS image is suitable for classification tasks such as vegetation cover estimation from satellite images (Gelvez et al., 2017). Note that this implies that the signal is not fully reconstructed but just a low-dimensional projection, reducing the complexity of dealing with HS images. This research thesis addresses the problem of recovering the Covariance Matrix of the signal and using it to guide the sensing process and reconstruct the signal in Compressive Spectral Imaging (CSI). Specifically, this thesis proposes designing the sensing protocol using the CM of hyperspectral imaging in an adaptative approach via a greedy binary algorithm. In CSI, the CM is unknown hence the CM recovery must also be addressed. CM estimation is addressed using a projected gradient algorithm with an Armijo search strategy to speed up the recovery. Additionally, this thesis proposes a compact optical architecture capable of capturing the random projections required for the CM estimation. Finally, a deep learning algorithm for land cover estimation is used with the proposed reconstruction methodology to test the correct performance in the application of HSI. The thesis organization is summarized as follows:

**Chapter 2**: Describes the proposed optimization problem and a greedy algorithm to design the sensing protocol of a compressive optical camera. The optimization problem seeks to maximize the quotient of binary patterns in the signal's subspace. That is, to maximize the product between the binary vectors and the scene covariance matrix. The optimization problem is solved via a greedy algorithm that finds a single vector at the time. The influence of the previous binary vectors is subtracted from the covariance matrix to find the next vector. Extensive simulations show that a few binary patterns can effectively preserve the covariance matrix's variance.

**Chapter 3**: Presents a convex optimization problem and a gradient descent algorithm to estimate the covariance matrix from random projections. The optimization problem uses a signal multi-partition set-up and a low-rank restriction to shrinkage the solution. The convergence guarantees and error terms are given. The problem is solved using a projected gradient descent algorithm using an Armijo search approach to speed up the convergence. Computational simulations and laboratory experiments show that the proposed strategy outperforms state-of-the-art algorithms based on second statistic moment estimation.

**Chapter 4**: Presents the implementation of the methods in chapter 3 for a video setup. The speed of the proposed approach enables the use of the proposed framework in a video setup. Hence this chapter proposes a new optical architecture and a modified optimization problem for a video set-up. The proposed architecture uses a lenslet array to capture multiple scene views simultaneously, avoiding the multishot requirement. Additionally, the optimization problem incorporates a low-rank restriction in the time dimension to take advantage of the temporal correlation of the scene. Experiments in the laboratory show that the proposed method can reconstruct around 8 frames per second and outperforms in terms of reconstruction quality to some state-of-the-art algorithms based on ADMM for image reconstruction.

**Chapter 5**: Addresses the land cover estimation problem from multispectral satellite images. For this use case, an area of $440 \times 680$ spatial pixels located at Valle de San José, Santander was selected. The image was acquired with the sentinel sensor in september 2022. The classification used 5 classes: grass, urban, forest, water, and other. The classes were validated via an in-situ visit in which multiple GPS points were acquired to generate an inventory of pixels for training.

The main objective is to validate the correct reconstruction of the image and the ability to classify correctly the information of the recovered images.

## Publications

The contributions of this dissertation have been led to the following publications

### International Journal Papers

1. **J. Monsalve**, H. Rueda-Chacon and H. Arguello, "Sensing Matrix Design for Compressive Spectral Imaging via Binary Principal Component Analysis," in IEEE Transactions on Image Processing, vol. 29, pp. 4003-4012, 2020, doi: 10.1109/TIP.2019.2959737.

2. **J. Monsalve**, J. Ramirez, I. Esnaola and H. Arguello, "Covariance Estimation From Compressive Data Partitions Using a Projected Gradient-Based Algorithm," in IEEE Transactions on Image Processing, vol. 31, pp. 4817-4827, 2022, doi: 10.1109/TIP.2022.3187285.

### International Conference Papers

1. Díaz, E., **J. Monsalve**, Guerrero, A., and Arguello, H. Covariance Matrix Estimation from Multiple Subsets in Compressive Spectral Imaging. Imaging and Applied Optics 2018 , OSA, Orlando, FL, USA, 2018, CTu5D.

2. Rojas, K., **J. Monsalve**, Gelvez, T., and Arguello, H. Correlation Matrix Estimation from Compressed Measurements in a Pattern Recognition System. Imaging and Applied Optics 2018, OSA, Orlando, FL, USA, 2018

3. **J. Monsalve**, M Marquez, I Esnaola, H Arguello Compressive Covariance Matrix Estimation from a Dual-Dispersive Coded Aperture Spectral Imager. IEEE International Conference on Image Processing (ICIP), Anchorage, Alaska, USA, 2021.

4. M Marquez, **J. Monsalve**, H Rueda, H Arguello Compressive spectral virtual multishot imager via lenslet array. Optical Sensors and Sensing Congress, Washington, DC, USA, 2021.

5. G Blanco, J Perez, **J. Monsalve**, M Marquez, I Esnaola, H Arguello Single Snapshot System for Compressive Covariance Matrix Estimation for Hyperspectral Imaging via Lenslet Array. XXIII Symposium on Image, Signal Processing and Artificial Vision (STSIVA), Popayán, Colombia, 2021.

6. **J. Monsalve**, M Marquez, I Esnaola, H Arguello Cocosvi: Single Snapshot Compressive Spectral Video Via Covariance Matrix Estimation. Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Rome, Italy, 2022

## Dissertation Overview

### Research question

How to improve the accuracy in the recovery of the first and second sample statistical moments from low-dimensional random projections and how to use these statistics to enhance the reconstruction of the data in compressive spectral imaging?

### Hypothesis

The first and second sample statistical moments of a dataset can be accurately estimated directly from low-dimensional random projections and they can be used to improve the reconstruction of the high dimensional data using compressive spectral imaging using principal component analysis based techniques.

### General objective

To design and optimize method to retrieve sample statistics from compressive spectral measurements preserving the original data's sub-space and to analyze the use of the sample statistics to reconstruct the underlying signal using compressive sensing theory.

#### Specific Objectives.

- To determine the most suitable sensing/projection protocols based on compressive sensing and random projections from the state-of-the-art applicable to hyperspectral imaging to be used in the statistics recovery.

- To design an algorithm based on the gradient descent method to recover the first and second sample statistical moments from low-dimensional random projections.

- To test the performance of the proposed algorithm to recover the sample statistics in hyperspectral imaging reconstruction.
- To adapt a state-of-the-art algorithm to estimate the vegetation cover using sample statistics and random low-dimensional projections of hyperspectral images based on the proposed approach.
- To verify the performance of the adapted algorithm comparing the accuracy in vegetation cover estimation with state-of-the-art algorithms.

# 1. Theoretical Background

## 1.1. General Notation

Throughout this thesis the following notation is used:

- $\mathbb{R}$ Set of real numbers.
- $\mathbb{R}^D$ $D$-dimensional linear space.
- The lowercase boldface letters denote vectors such as $\mathbf{x} \in \mathbb{R}^D$.
- Uppercase boldface letters denote matrices, such as $\mathbf{X} \in \mathbb{R}^{D \times N}$.
- The transpose of the matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ is denoted as $\mathbf{X}^T \in \mathbb{R}^{N \times D}$.
- $\|.\|_p$ represents the $\ell_p$ norm for $p > 0$.
- $\|.\|_F$ represents the Frobenius norm.
- $\mathbb{E}[.]$ is the mathematical expectation.

## 1.2. Spectral imaging

Spectroscopy is a powerful technique that enables material identification based on its spectral response along the electromagnetic spectrum, by capturing the spectral response of a single spatial position (Bioucas-Dias et al., 2013). When a light source illuminates a target, the reflected photons are measured, and since each material absorbs and reflects different wavelengths, the resulting spectral signature is unique for each material. This signature comprises hundreds of narrow spectral bands, making spectroscopy highly effective when analyzing a few spots, particularly in

scenes with homogeneous spatial distribution.

On the other hand, traditional RGB imaging captures only three central wavelengths for each spatial position, providing a more accurate spatial map distribution of the scene, but far less spectral information. Consequently, RGB imaging is more appropriate for shape-based classification, but fails when classifying a single spatial position or irregular objects.

To address this issue, spectral imaging has emerged as a solution, acquiring dense spatial and spectral information in a 3-dimensional image known as a datacube (Manolakis et al., 2003). These images are known as multispectral images (MS) or hyperspectral images (HS), depending on the number of acquired spectral bands, and enable material identification since each material has its unique spectral signature. The concept of spectral imaging is illustrated in Fig. 1.

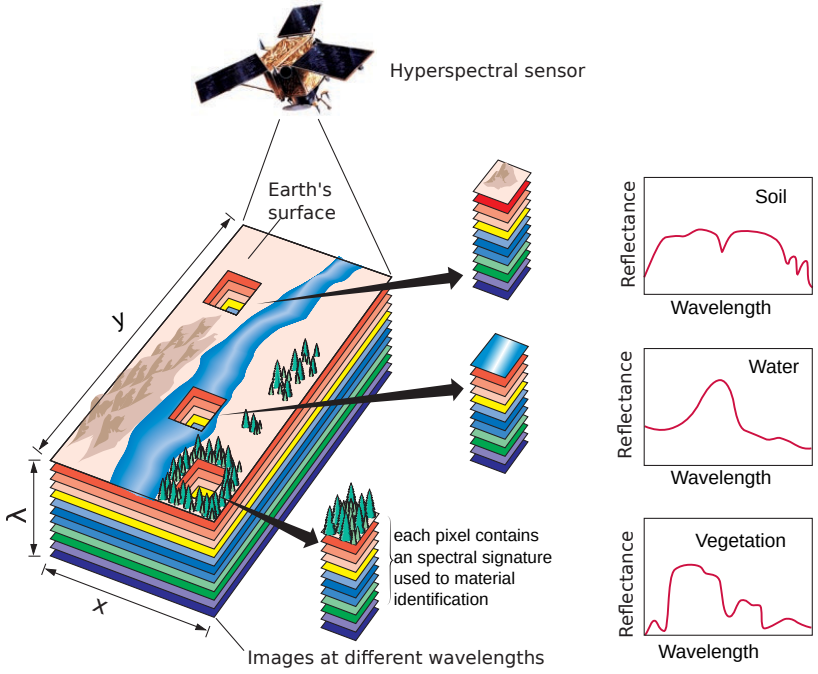


*Figure 1.* Schematic representation of a spectral image. Each pixel contains information along the electromagnetic spectrum used for material identification.

The number of spectral bands is related to the classification specificity. For instance, tens of bands are enough to differentiate between green paint and trees, while hundreds of bands make possible even tree species classification. Nevertheless, one important difficulty in hyperspectral imaging acquisition is that cameras can only acquire 2-dimensional images at each time. Since a hyperspectral image contains 3 dimensions, it can not be captured in a single shot. Different approaches have been proposed to acquire this type of image (Lu and Fei, 2014).

- **Point scanning**: Point scanning methods acquire a single spatial position each time along the spectrum. At the end of the process, the captured points are concatenated to form the 3D image, see Fig. 2 A). This is one of the most time-consuming approaches, but the spectral resolution is usually higher (Fun, 2022).

- **Spatial scanning**: This approach is also known as push-broom since it scans a whole spatial line along the spectral dimension, See Fig. 2 B). This approach is the most common in remote sensing applications since the platform where the camera is transported is moving (Sousa et al., 2022).

- **Spectral scanning**: Spectral scanning methods use color filters to acquire the whole image in a single spectral band; the number of spectral bands that can be captured depends on the number of filters and their spectral response. This approach is sketched in Fig. 2 C) (Foley et al., 2022).

- **Snapshot acquisition**: This approach is a 2D projection of the 3D image; for that reason, a reconstruction step is needed. However, it does not require a scanning process resulting in

less acquisition time (see Fig. 2 D) (Arce et al., 2013).

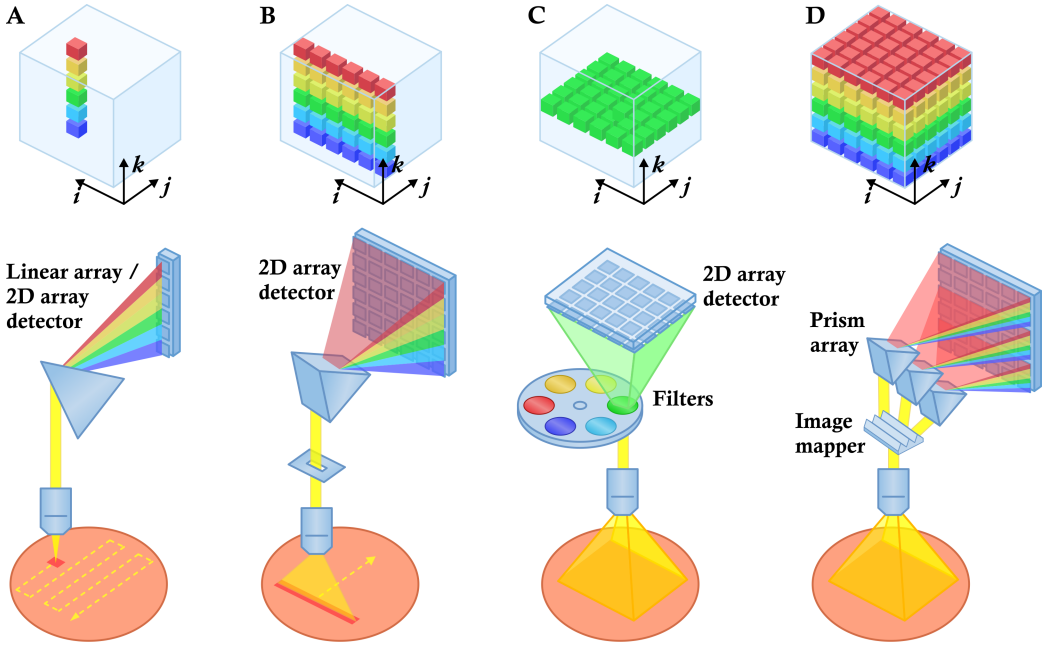


*Figure 2.* Schematic representation of the different approaches to acquire a hyperspectral image. a) Point scanning, b) Spatial scanning, c) Spectral scanning, d) Snapshot acquisition .

Hyperspectral imaging has been successfully used in many areas such as precision farming (Candiago et al., 2015), (Xiong et al., 2015), medical diagnosis (Levenson and Mansfield, 2006), and others (Markas and Reif, 1993; Manolakis et al., 2003). Nevertheless, the huge dimensions of the resulting datacube in HS imaging come with challenges related to its storage, transmission, and processing. To mitigate these problems, many approaches are based on lossy compression algorithms that require capturing the full image to compress it. However, this procedure does not solve the problem but transfers the bottleneck from the acquisition, transmission, and storage to the computation field. For that reason, approaches that compress the signal while it is acquired are vital to solving these inconveniences.

### 1.3. Compressive sensing and random projections

Random projections consists in projecting/compressing a signal $\mathbf{f} \in \mathbb{R}^l$ onto a random low-dimensional subspace given by

$$\mathbf{y} = \mathbf{H}\mathbf{f} + \mathbf{n}, \tag{1}$$

where $\mathbf{H} \in \mathbb{R}^{m \times l}$ is the sensing/projection matrix, with $m < l$ and $\mathbf{n} \in \mathbb{R}^l$ is additive noise. Since (1) is underdetermined estimating $\mathbf{f}$ from $\mathbf{y}$ is a challenging problem (Arce et al., 2014a; Candes and Wakin, 2008; Fowler, 2009). One of the most traditional ways to recover $\mathbf{f}$ is by using the least squares approach, which consists in solving the optimization problem (Cline and Plemmons, 1976)

$$\begin{aligned} \mathbf{f}^* = \underset{\mathbf{f}}{\text{arg. min}} \quad & \|\mathbf{f}\|_2^2 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{H}\mathbf{f}, \end{aligned} \tag{2}$$

where $\|.\|_2^2$ represents the $\ell_2$ norm. An advantage of (1) is that it can be solved via a closed-form as $\tilde{\mathbf{f}} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{y}$, nevertheless, the mean squared error of the estimated signal is usually large, especially when the compression increases, i.e., $m \ll n$ (Romero et al., 2016a).

On the other hand, Compressive sensing is a theory that dictates that a signal can be accurately reconstructed from a set of low-dimensional random projections if the signal admits a sparse or low-rank representation in some domain (Arce et al., 2014a; Candes and Wakin, 2008; Galvis-Carreño et al., 2014; Mojica et al., 2017). Let $\mathbf{\Psi} \in \mathbb{R}^{l \times l}$ be an orthonormal basis such as discrete cosine or wavelet basis. Then, a signal $\mathbf{f}$ is said to be s-sparse in $\mathbf{\Psi}$ domain if its representation $\boldsymbol{\theta} = \mathbf{\Psi}\mathbf{f}$ contains at much $s$ non-zero coefficient. Using this concept, (1) can be reformulated to

take into account the orthonormal basis as

$$\mathbf{y} = \mathbf{H}\boldsymbol{\Psi}^T\boldsymbol{\theta} + \mathbf{n}, \tag{3}$$

using this sparsity property, the recovery problem (2) can be reformulated to promote the signal to be sparse in the basis domain

$$\begin{aligned} \boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{arg. min}} \quad & ||\boldsymbol{\theta}||_0 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{H}\boldsymbol{\Psi}^T\boldsymbol{\theta}, \end{aligned} \tag{4}$$

where $||.||_0$ is the $\ell_0$ norm. However, (4) has two problems, first is that the restriction $\mathbf{y} = \mathbf{H}\boldsymbol{\Psi}^T\boldsymbol{\theta}$ could never be satisfied due to the noise, and second it is an NP-hard problem because it requires to calculate an $\ell_0$ norm (Candes and Tao, 2005). For that reason, (4) is usually relaxed to an optimization problem known as Basis Pursuit, which uses $\ell_1$ norm instead (Candes and Tao, 2005; Liu and Zhang, 2014) and is formulated as

$$\begin{aligned} \boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\text{arg. min}} \quad & g(\boldsymbol{\theta}) \\ \text{subject to} \quad & \mathbf{y} - \mathbf{H}\boldsymbol{\Psi}^T\boldsymbol{\theta} < \varepsilon. \end{aligned} \tag{5}$$

where $g(.)$ is a function that promote sparsity through the $\ell_1$ norm, low-rank through trace, or smoothness trhough total variation; and $\varepsilon$ is an error term. Further (5) can be cast to an equivalent

problem called lasso as

$$\boldsymbol{\theta}^* = \arg.\min_{\boldsymbol{\theta}} \quad ||\mathbf{y} - \mathbf{H}\boldsymbol{\Psi}^T\boldsymbol{\theta}||_2^2 + \lambda g(\boldsymbol{\theta}), \tag{6}$$

(6) can be solved if $\boldsymbol{\theta}$ is sparse and if the sensing matrix is incoherent with the domain basis $\mathbf{H}$.

**1.3.1. Incoherence.** Incoherence is an important property in the compressive sensing area since they give an idea of the reconstruction probability of a signal from compressed measurements (Arce et al., 2014a; Candes and Wakin, 2008).

The coherence $\mu$ is defined as the maximum correlation that exists between any of the atoms of the sensing matrix $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_m]^T$ and the representation basis $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, \cdots, \boldsymbol{\psi}_l]$. Mathematically, it can be expressed as (Candes and Wakin, 2008)

$$\mu(\mathbf{H}, \boldsymbol{\Psi}) = \sqrt{(}n) \max_{1 \leq m,j \leq l} \quad |\mathbf{h}_k^T \boldsymbol{\psi}_j|. \tag{7}$$

It has been proved that the amount of measurements needed to reconstruct the signal is given by (Candès and Romberg, 2007)

$$m \geq C\mu(\mathbf{H}, \boldsymbol{\Psi})^2 s \log(l), \tag{8}$$

Where $C$ is a constant. Hence, to reduce the number of measurements, the sparsity $s$ and the coherence must be as small as possible, i.e. the signal must be sparse, and the sensing matrix must be incoherent with the representation basis.

**1.3.2. Restricted isometry property .** Another important property is the restricted isometric property (RIP). RIP measures the behavior of a matrix as an orthonormal system when sparse combinations are used (Candes and Tao, 2005). This ensures that the pairwise distances of sparse vectors are preserved under the low-dimensional projection. Mathematically, the RIP for an S-sparse vector is defined as the smallest constant $\delta_s$ such that

$$(1-\delta_s)||\boldsymbol{\theta}||_2^2 \leq ||\mathbf{H}\boldsymbol{\Psi}^T\boldsymbol{\theta}||_2^2 \leq (1+\delta_s)||\boldsymbol{\theta}||_2^2. \tag{9}$$

The signal's reconstruction probability increases as $\delta_s$ decreases (Pinilla et al., 2016). Hence, based on these two properties RIP and incoherence, multiple sensing matrix designs have been proposed. However, the signal reconstruction can fail if the signal is not sparse on a given representation basis. Hence, another important aspect is constructing a representation basis where the data is sparse. Universal representation basis such as DCT and Wavelet promotes sparsity in many natural scenes. Nevertheless, a better representation basis can be constructed by considering the signal's statistical information. This problem is known as dictionary learning.

**1.3.3. Processing on random projection.** Another commonly used approach in random projection is to process the data directly in the compressed domain. Many algorithms have been proposed to work with random projections in applications such as clustering, and deep learning Hinojosa et al. (2018a). Similarly to RIP in CS, theoretical results support using random projections. One of the most used results is known as Johnson–Lindenstrauss lemma (Johnson and Lindenstrauss, 1984; Dasgupta and Gupta, 2003), which basically states that for any high

dimensional data set there exist a mapper to a lower-dimensional subspace such that the pairwise distances are preserved. Mathematically, for any set $[\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_p] \in \mathbb{R}^l$ there exists a map function $g : \mathbb{R}^l \longrightarrow \mathbb{R}^m$ such that

$$(1-\varepsilon)||\mathbf{f}_i - \mathbf{f}_j||_2^2 \leq ||g(\mathbf{f}_i) - g(\mathbf{f}_j)||_2^2 \leq (1+\varepsilon)||\mathbf{f}_i - \mathbf{f}_j||_2^2, \tag{10}$$

for $\varepsilon > 0$, and $i, j = \{1, 2, \cdots, p\}$. In contrast with to RIP, the JL lemma does not impose any sparsity restriction on the data, allowing it to work with the data directly in the compressed domain.

### 1.4. Compressive covariance sampling

In many fields, reconstructing the signal is not the main goal, instead, statistical information is more relevant. Thus, the compressive covariance matrix (CCS) surged as a solution. The covariance matrix (CM) plays a central role in spectral imaging applications such as dictionary learning (Rubinstein et al., 2012), dimensionality reduction (Van Der Maaten et al., 2009), and classification among others (Romero et al., 2016a; Bioucas-Dias et al., 2014; Mohammadi et al., 2014). Conventional CM estimators require full knowledge of the spectral image, and therefore, traditional spectral imagers require high communication, storage, and processing capabilities. To overcome these challenges, recent approaches have explored the use of Compressive Spectral Imaging (CSI) theory (Foucart and Rauhut, 2013; Arce et al., 2013). Because CSI imagers acquire a compressed version of the spectral image (Marquez et al., 2020b, 2019, 2017), conventional CM estimators require the reconstruction of the signal. Compressive Covariance Sampling (CCS) has emerged as an alternative to estimate the covariance matrix, directly from the compressive measurements to

avoid this costly reconstruction step. Although several strategies related to second-order statistics estimation have been proposed (Bioucas-Dias et al., 2014; Fowler, 2009), real optical implementations have not been considered.

**1.4.1. Unbiased covariance estimator:.** The covariance matrix is a statistical measure that is theoretically computed using the expectation operator. However, in real-world scenarios where the number of samples is limited, it is not feasible to calculate it this way. Therefore, various estimators have been developed to compute this statistic under limited sample sizes. Depending on the situation, one estimator may be considered better than another if it is unbiased.

Let $\boldsymbol{X}$ be a random vector in $\mathbb{R}^l$, the first-statistic moment of the random vector is given by

$$\boldsymbol{\mu} = \mathbb{E}[\boldsymbol{X}], \tag{11}$$

The covariance matrix is given by

$$\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})^T], \tag{12}$$

where $\mathbb{E}$ represents the mathematical expectation. Given a set of realizations $\mathbf{x}_i$ of a random vector $\boldsymbol{X} \in \mathbb{R}^l$ $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, the maximum likelihood estimator for the mean is given by

$$\tilde{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \tag{13}$$

and for the covariance matrix is

$$\mathbf{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \tilde{\mathbf{x}})(\mathbf{x}_i - \tilde{\mathbf{x}})^T, \tag{14}$$

both $\tilde{\mathbf{x}}, \mathbf{S}$ are unbiased estimators for $\boldsymbol{\mu}, \boldsymbol{\Sigma}$. The later means that $\boldsymbol{\mu} = \mathbb{E}[\tilde{\mathbf{x}}]$ and $\boldsymbol{\Sigma} = \mathbb{E}[\mathbf{S}]$. The problem of finding an unbiased estimator for the mean and the covariance matrix becomes more challenging when the random vector is projected in a random subspace as

$$\mathbf{Y} = \mathbf{P}^T\mathbf{X} + \mathbf{N}, \tag{15}$$

where $\mathbf{P} \in \mathbb{R}^{l\times m}, m < l$ is the projection matrix, and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{l\times n}$ are the realization of the random vector. In this scenario, the unbiased estimator for the mean is given by (Qi and Hughes, 2012; Pourkamali-Anaraki, 2016)

$$\hat{\mathbf{x}} = \frac{l}{mn}\sum_{i=1}^{n}\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{y}_i. \tag{16}$$

For the covariance matrix, finding an unbiased estimator is more challenging; an intuitive estimator (assuming zero mean) is given by

$$\mathbf{C}_p = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{y}_i)(\mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{y}_i)^T. \tag{17}$$

However, it should be noted that (17) is a biased estimator, but the eigenvectors are preserved (Qi

and Hughes, 2012; Pourkamali-Anaraki, 2016). In fact, the sample covariance matrix $\mathbf{S} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T$ is related to $\mathbf{C}_p$ as

$$\lim_{n\to\infty} \mathbf{C}_p = \mathbf{W}\mathbf{C}\mathbf{W}^T + \frac{m}{l}\varepsilon^2\mathbf{I}, \tag{18}$$

where $\mathbf{C}$ is a diagonal matrix

$$\mathbf{C} = diag\Big(\lambda_1 k_1 + \frac{\sum_{j=1, j\neq 1}^{d} \lambda_j^2 k_2}{l-1}, \ldots, \lambda_d^2 k_1 + \frac{\sum_{j=1, j\neq d}^{d} \lambda_j^2 k_2}{l-1}, \frac{\sum_{j=1, j\neq d}^{d} \lambda_j^2 k_2}{l-1}, \ldots\Big), \tag{19}$$

where $\lambda_i$ are the eigenvalues of $\mathbf{S}$, $k_1 = m^2/l^2 + 2m(l-m)/(l^3+2l^2)$ and $k_2 = m/l - k_1$. Hence, (17) is a biased estimator of $\mathbf{\Sigma}$, but it preserves its eigenvectors as long as $n \to \infty$.

**1.4.2. Compressive-Projection Principal Component Analysis.** Compressive-Projection Principal Component Analysis (CPPCA) is a technique to recover second-order moment information from random projection. In contrast to (17), this method does not require an infinite number of samples. This method uses the Rayleigh-Ritz theory to bound the angle between the eigenvectors of the sample covariance matrix and the compressed version of the covariance matrix (Jia and Stewart, 2000). Let

$$\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^T, \tag{20}$$

be the sample covariance matrix, and

$$\tilde{\mathbf{S}} = \frac{1}{n}\mathbf{Y}\mathbf{Y}^T = \frac{1}{n}\mathbf{P}^T\mathbf{X}\mathbf{X}^T\mathbf{P}, \tag{21}$$

be the compressed version of the covariance matrix. the eigen-decomposition of (20) and (21) are respectively $\mathbf{S} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T \in \mathbb{R}^{l\times l}$ and $\tilde{\mathbf{S}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^T \in \mathbb{R}^{m\times m}$. Given that $\mathbf{W}$ and $\tilde{\mathbf{U}}$ have different dimensions, let us define $\mathbf{U} = \mathbf{P}\tilde{\mathbf{U}}$ as the projection onto the subspace of $\mathbf{P}$. Additionally, let $\mathbf{v}_i = \mathbf{P}\mathbf{P}^T\mathbf{w}_i/\|\mathbf{P}\mathbf{P}^T\mathbf{w}_i\|_2^2$. Defining $\phi_n$ as the angle between $\mathbf{w}_n$ and $\mathbf{v}_n$; and $\theta_k$ as the angle between $\mathbf{u}_k$ and $\mathbf{w}_n$ it holds that (Fowler, 2009)

$$|\sin\phi_n| \leq |\sin\theta_k| \leq \frac{\|\mathbf{S}\mathbf{u}_k - \mathbf{u}_k\mathbf{u}_k^T\mathbf{S}\mathbf{U}_k\|_2}{\gamma_k}, \tag{22}$$

where $\gamma_k$ is the minimum separation between two eigenvalues; i.e., the difference between the estimated eigenvector $\mathbf{u}_k$ and an eigenvector $\mathbf{w}_n$ depends on the separation of the corresponding eigenvalue with the rest of eigenvalues. To decrease this angle, authors propose to divide the signal into $J$ subsets $\mathbf{X}_i \in \mathbb{R}^{l\times n/J}$ and project it using its own matrix $\mathbf{P}_i$. The intuition with this partition setup is that the sample covariance matrix of each partition $\mathbf{S}_i$ approximates the sample covariance matrix $\mathbf{S}$. Hence the intersection between the recovered eigenvectors $\mathbf{u}_k$ for each partition will be closer to $\mathbf{w}_n$ as long as the separation condition holds. For that, let us construct a projector

$$\mathrm{Q} = \mathrm{P}^{\perp} \oplus \mathrm{span}\{\mathbf{v}\}, \tag{23}$$

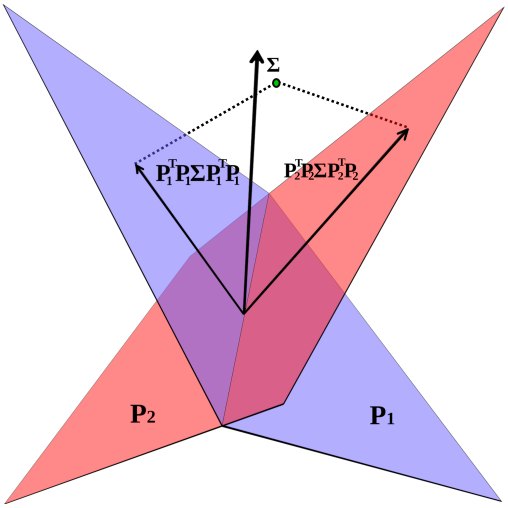


*Figure 3.* The recovered eigenvector is the intersection of the recovered eigenvectors onto the different subspaces $\mathbf{P}_i$.

where $P^{\perp}$ is the orthogonal complement of $\mathbf{P}$; then, the resulting Q contains the null space of $\mathbf{P}$ and the 1-D space of the projection $\mathbf{v}$. The intersection of each recovered eigenvector is done via

$$\tilde{\mathbf{w}}_i = \frac{1}{J} \sum_{j=1}^{J} \mathbf{Q}_j \mathbf{Q}_j^T \tilde{\mathbf{w}}_{(i-1)}. \tag{24}$$

Equation (24) averages the J eigenvectors associated with each partition. The intuition of (24) is shown in Fig. 3.

### 1.5. Deep learning in hyperspectral imaging

Traditionally, inverse problems have been solved via a model-based approach; in this scenario, a mathematical model of the observations is proposed and solved using an optimization algorithm that penalizes a loss function. However, the main problem with this approach is that the model usually simplifies the real observation. Model simplification is important because the

resulting problem could be impossible to solve if the model is too complex. The deep learning approach uses a setup that trusts more in the data than in the mathematical model. Hence, many deep learning algorithms aim to minimize the expectation of a loss function with respect to a known dataset. The most common deep-learning algorithms in imaging are convolutional neural networks (CNN) (Arguello et al., 2021; Monroy et al., 2022; Ramirez et al., 2021; Wang et al., 2022). This kind of algorithm uses kernels that perform multiple convolutions in different scales of the image extracting the key features of the images, allowing multiple tasks such as classification (Redmon and Farhadi, 2018), reconstruction (Monroy et al., 2022; Arguello et al., 2021), denoising (Xu et al., 2015), segmentation (Ronneberger et al., 2015), among others. The CNN algorithms for inverse problems can be classified into three types depending on the strategy used to solve the problem: supervised, unsupervised, and plug-and-play. More methods can be classified as a mixture of supervised and unsupervised. For instance, self-supervised, internal learning, fine-tuning, and transfer learning, among others.

**1.5.1. Supervised CNN.** This approach requires a full dataset containing the measured image and the solution or classification, usually known as ground truth. For instance, in a segmentation problem, the captured images and their respective segmentations are required. The training consists of showing the image and the expected response to the network, in order to minimize the error with respect to the solution of the CNN model. The CNN model is usually made up of convolutional, non-linear, fully connected, and regularization layers. There exist a lot of models that have a general purpose, meaning they can be used for many problems (Ronneberger et al., 2015; Simonyan and Zisserman, 2014). The specific application is given to the model in the

refining and training step.

***1.5.1.1. CNN models for classification.*** This was, perhaps, the first task to be addressed using CNN models. The classification consists in assigning a label to an image related to its content/concept. The classification concept in CNN is shown in Fig. 4-a). The first CNN successfully implemented for classification was Alexnet (Krizhevsky et al., 2017). Alexnet was a CNN model that require a huge number of learnable parameters. For that reason, many subsequent works focused on reducing the number of parameters while improving the accuracy. VGG net was proposed as a very deep CNN that reduces the number of parameters, mainly decreasing the number of fully connected layers and using more convolutional layers to reduce the image size (Simonyan and Zisserman, 2014). Inception v3 used a 1-D kernel strategy to reduce the number of parameters even more. This strategy uses two filters with dimensions $1 \times 3$ and $3 \times 1$, in contrast to a single kernel $3 \times 3$ reducing from 9 to 6, the number of learnable parameters for a window (Szegedy et al., 2016). Lightweight CNN models that can be executed in computationally limited devices have also been proposed. These models usually have less accuracy, but the number of parameters allows them to be executed on mobile platforms. An example of this kind of model is MobileNet (Howard et al., 2017).

***1.5.1.2. CNN models for object detection.*** A more challenging task is object detection. It consists of classifying an image's content and locating it inside the image using a bounding box. This computer vision technology has been used to detect generic objects such as houses, cars, animals, etc (Redmon and Farhadi, 2018). Also, as part of the biometric applications by locating faces (Serengil and Ozpinar, 2021). There exist two types of object detection algorithms; single-

stage and multi-stage. The multi-stage algorithm detects possible objects, and then a classifier assigns a label to the detected objects. The most popular multi-stage CNN model is RCNN and its variations (Girshick et al., 2013; Girshick, 2015). A schematic representation of this technique is shown in Fig. 4-b). The single-stage algorithms require only one step for the detection and classification. The most popular algorithm is Yolov3 (Redmon and Farhadi, 2018). However, multiple versions have been published after the original Yolo work (Bochkovskiy et al., 2020; Li et al., 2022).

***1.5.1.3. CNN models for segmentation:.*** Segmentation is a higher-level classification task consisting of labeling every pixel in an image with a concept. This type of classification is more accurate than object detection since the classification must be exact in a pixel-wise manner. Unet is the most common CNN architecture used for this task (Wang et al., 2022; Ronneberger et al., 2015). A representation of this type of task is depicted in Fig. 4-c).

***1.5.1.4. CNN models for regression.*** In imaging, regression consists in recovering a signal from a set of degraded measurements. For that, let $\mathbf{x}$ be an image and $f$ an operator corresponding to a degradation to the image. Hence the measurements are given by

$$\mathbf{y} = f(\mathbf{x}) + \mathbf{n}, \tag{25}$$

where $\mathbf{n}$ represents stationary additive Gaussian noise. The objective of a regression algorithm is to estimate the relation between the measurements $\mathbf{y}$ and any information related to $\mathbf{x}$. Traditional model-based methods start by finding an inverse operator of $f$. In contrast, most CNN approaches

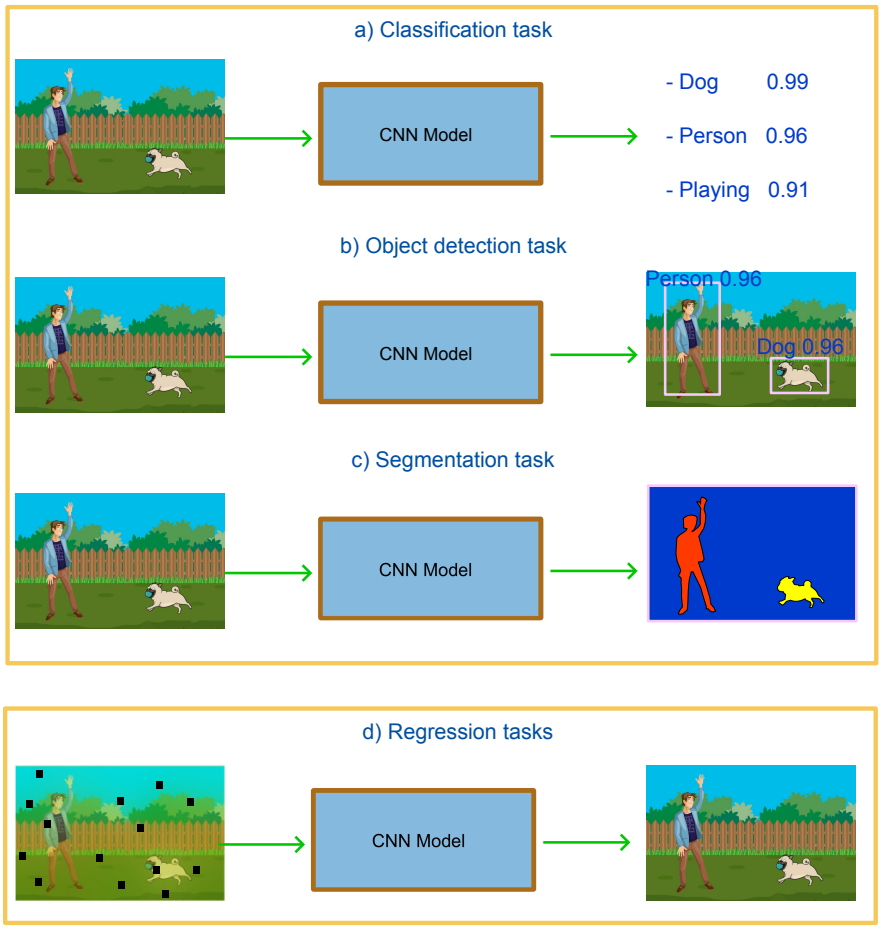


*Figure 4.* Schematic of the different types of tasks performed by CNN and the differences between them.

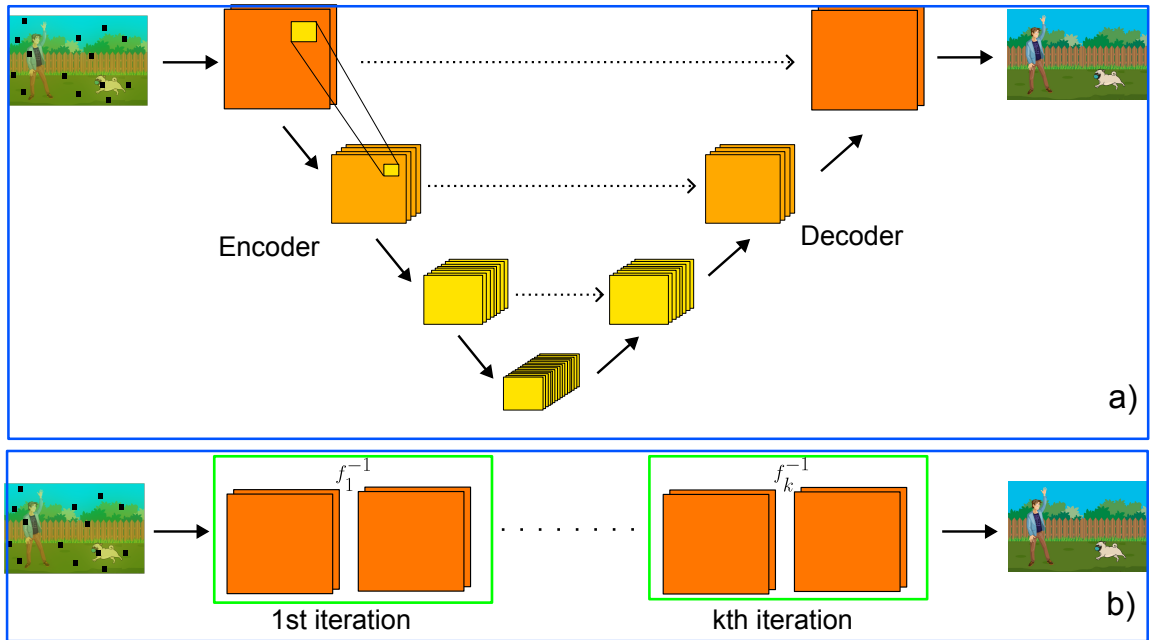


*Figure 5.* Schematic representation of CNN models for inverse problems. a) representation of a supervised encoder-decoder architecture. b) representation of an unrolling method

do not consider the operator $f$. The approach consist in minimizing a loss given by

$$\min_\theta L(g_\theta(\mathbf{y}) - \mathbf{x}), \tag{26}$$

where $g_\theta$ represents the convolutional model, and $L$ is any differentiable penalty function. In the training process, the degraded signal $\mathbf{y}$ and the true signal $\mathbf{x}$ are required to update the weights $\theta$ such that (26) is minimized.

Most popular CNN models to perform regression are based on an encoder-decoder architecture. This architecture performs multiple analyses on different scales, down-sampling the features on the encoder side and up-sampling the features on the decoder side. This concept is depicted in Fig. 5. Unet is a common architecture that uses this approach, which is not only used for seg-

mentation tasks (Miao et al., 2019; Xiao et al., 2022). However, encoder-decoder models do not use any intuition of an inverse problem in the solution and only want to minimize the loss. Another method for inverse problems using deep learning takes advantage of the existing literature in convex optimization. For that, the neural network is modeled as multiple layers representing iterations of a traditional algorithm such as an ADMM or gradient descent. This method is known as unrolling algorithms (Khader et al., 2022; Marquez et al., 2022; Zhao et al., 2018).

**1.5.2. Unsupervised CNN.** This deep learning approach could require a single image or a whole dataset, but the labels or respective classifications are not required. The loss function of these methods only depends on the known unlabeled data. Depending on the application, different types of neural networks exist for unsupervised tasks.

***1.5.2.1. Autencoders.*** Autoencoders are a powerful technique to learn data characteristics by reducing its dimensionality. This architecture comprises two parametrized functions $E_{\theta_1}, D_{\theta_2}$. The encoder function takes as input an image $\mathbf{x} \in \mathbb{R}^n$ and returns a coded version $\mathbf{y} \in \mathbb{R}^m$; usually, the dimension of the coded version is smaller than the input signal, i.e., $m < n$ (Tao et al., 2015; Zhou et al., 2019). The encoder depends on a set of parameters $\theta_1$ learned during the training stage. The decoder function performs the inverse operation and recovers an approximation of the image, $D_{\theta_2}(\mathbf{y}) = \mathbf{x}$. Hence, the autoencoder is trained in an end-to-end manner minimizing the loss that usually looks like

$$\min._{\theta_1,\theta_2} L(D_{\theta_1}(E_{\theta_2}(\mathbf{x})) - \mathbf{x}), \tag{27}$$

note that the loss only depends on the known data **x**. Autoencoders are useful in feature extraction and dimensionality reduction since they learn to convert an input into a compact version by preserving important information.

***1.5.2.2. Deep prior.*** Deep prior (DP) is a type of neural network used in imaging inverse problems, that assumes that neural networks can learn the posterior distribution of an image. Hence, DP can reconstruct an image from a set of degraded measurements. DP only requires the degraded signal and a mathematical degradation model to perform the training. In contrast to other CNN methods, DP is only suitable for a single image, i.e. ,the training consists in learning the posterior distribution of a single image (Lin et al., 2020; Monroy et al., 2022). This type of neural network (NN) minimizes the loss function

$$\min._{\theta_1} L(H(f_\theta(\mathbf{y})) - \mathbf{y}), \tag{28}$$

where $H$ is the mathematical degradation model and $f_\theta$ is the neural network.

***1.5.2.3. Generative adversarial neural networks.*** Generative adversarial neural networks (GAN's) are a type of NN intended for data generation. GANs learn the data distribution from a training dataset and can generate real-looking data. The data generator is a neural network known as a generator $G_\theta$. The generator learns to create new data to make it look real, while another network, known as a discriminator $D_\theta$, tries to identify real and generated samples. This is an unsupervised method since the data does not need a label. The loss function minimized is given

by

$$\min_G \max_D \ \log D_\theta(\mathbf{x}) + \log(1 - D_\theta(G_\theta(\mathbf{z}))) \tag{29}$$

Although this type of network provides realistic results, it is also very unstable in the training stage, and some modifications have been proposed to solve the problem such as use change the discriminator function $D = [0,1]$ for a function $E = [0,\infty]$.

# 2. Binary Sensing Protocol Design using the Covariance Matrix of the Signal

*This chapter addresses the first objective of the thesis:*

- *To determine the most suitable sensing/projection protocols based on compressive sensing and random projections from the state-of-the-art applicable to hyperspectral imaging to be used in the statistics recovery.*

## 2.1. Introduction

Compressive spectral imaging (CSI) is a framework to acquire and compress spectral images by means of coded bi-dimensional projections, such that, the number of required measurements for reconstruction are fewer than those needed by traditional techniques based on the Shannon-Nyquist sampling theorem (Arce et al., 2014b; Cao et al., 2016; Candes and Wakin, 2008; Candes and Tao, 2005). CSI exploits the fact that natural scenes can be accurately represented in a lower dimensional subspace. This concept is known as sparsity or low rank behavior (Candes and Wakin, 2008; Arguello and Arce, 2012a; Donoho, 2006; Fowler, 2009). Further, the linear projector, so-called sensing matrix, used to capture the compressed version of the spectral image has to be incoherent with the representation basis, where the data becomes sparse. This, in turn, guarantees with high probability an accurate reconstruction, since an incoherent matrix has a dense representation in the basis domain, and so, no assumption on the behavior of the data is required.

Principal component analysis (PCA) is a technique used to reduce the dimensionality of a signal by projecting it into a lower dimension, such that, most of its variance is explained (Fowler,

2009; Qi and Hughes, 2012; Zhang et al., 2012). In particular for spectral images, PCA projects the spectral data using the eigenvectors associated with the $m$ greatest eigenvalues of the covariance matrix $\mathbf{\Sigma}$, resulting of the signal $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_n] \in \mathbb{R}^{l \times n}$, where $l$ is the number of spectral bands, $n$ the number of spatial pixels, and $\mathbf{f}_i \in \mathbb{R}^l$ is a pixel, for $i = 1, \cdots, n$. Thus, a matrix $\mathbf{W}_m \in \mathbb{R}^{l \times m}$, with the $m$ eigenvectors as columns, is used to project the data as $\tilde{\mathbf{F}} = \mathbf{W}_m^T \mathbf{F}$, with $\tilde{\mathbf{F}} \in \mathbb{R}^{m \times n}$ and $m < l$. This formulation has shown to achieve a small error in the Euclidean sense, described by $||\mathbf{F} - \mathbf{W}_m \tilde{\mathbf{F}}||$, while preserving the structure of the data in the low-dimensional space, and thus the direction of greatest variability (Kwak, 2008).

In a similar way, the noiseless CSI sensing procedure can be expressed as $\mathbf{Y} = \mathbf{Q}^T \mathbf{F}$, where $\mathbf{Q} \in \mathbb{R}^{l \times m}$ is the sensing matrix and $\mathbf{F}$ is the input image (Fowler, 2009; Bioucas-Dias et al., 2014a). CSI can be categorized as a dimensionality reduction technique, since it projects the spectral signal in a low-dimensional subspace spanned by the rows of the sensing matrix $\mathbf{Q}$. Note however that, $\mathbf{Q}$ is either randomly generated or designed based on, the restricted isometry property (RIP) or its incoherence with a representation basis (Correa et al., 2016a; Rueda et al., 2016; Correa et al., 2016d; Lin et al., 2014a). In other words, $\mathbf{Q}$ does not rely, conventionally, on the input signal. Therefore, much effort has been done in the signal processing community to design $\mathbf{Q}$ such that, the structure of the data is preserved in the low-dimensional subspace. Figure 6 shows an example of how PCA better preserves the direction of greatest variability of the data compared to random matrices. In this figure a dataset in $\mathbb{R}^3$ is projected onto a $\mathbb{R}^2$ subspace using the eigenvectors associated with the covariance matrix of the data and compared against a conventional random matrix. Note that the PCA projection better preserves the data separability in the $\mathbb{R}^2$ subspace

whereas the random projection mixes them all.

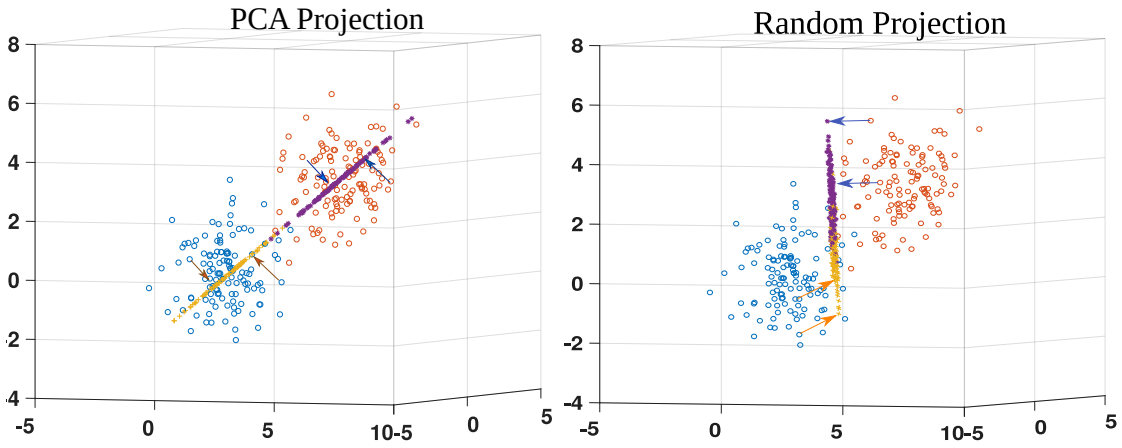


*Figure 6.* Example of how PCA preserves the structure of the data. Blue data points are projected to the yellow ones, and orange data points are projected to purple ones. (Left) Data in $\mathbb{R}^3$ projected onto a subspace in $\mathbb{R}^2$ using 2 eigenvectors, where the separability and direction of greatest variability of data is preserved. (Right) The same data is projected using a random matrix.

Remark that, if the CSI sensing matrix $\mathbf{Q}$ is equal to the PCA matrix $\mathbf{W}_m$, the compression or low-dimensional projection, can be considered optimal in the least squares sense (Bioucas-Dias and Nascimento, 2008). However, PCA is data-dependent, which requires to know the spectral image to be compressed beforehand, thus prohibiting its usage in CSI, where the target data are unknown a priori. Nevertheless, important information about the spectral data can be extracted from some random compressed measurements $\mathbf{Y}$, so that, an approximation of the covariance matrix of $\mathbf{F}$ can be attained, and thus exploited to design the subsequent sensing matrices via PCA. Another problem that appears in the design of $\mathbf{Q}$ via PCA is that the entries of $\mathbf{Q}$ are usually binary, as required by CSI optical architectures, while the entries of the principal component matrix $\mathbf{W}_m$ are usually real. Therefore, the goal of this work is to use the PCA intuition to design binary matrices $\mathbf{Q}$ that span a subspace where most of the variance of the signal $\mathbf{F}$ is explained. More

precisely, we aim to approximate the behavior of $\mathbf{W}_m$ with a binary matrix $\mathbf{Q}$ to be used as the CSI sensing matrix. For that reason, the traditional optimization problem, to find the eigenvectors, is modified by adding a binary constraint, and a computational algorithm is proposed to solve it efficiently.

**2.1.1. Related Work.** The use of PCA in compressive sensing has been previously studied to improve the signal reconstruction quality. For instance, Masiero *et al.*, in (Quer et al., 2012), used PCA to choose the best representation basis in the reconstruction process, whereas, Ke *et al.* used PCA to design sensing matrices for low-light-level imaging ($L^3$-imaging) (Ke and Lam, 2016) and feature-specific imaging (Ke et al., 2010). Although in (Ke et al., 2010) the design is adaptively obtained, the resulting vectors are chosen from a fixed basis like Hadamard. The ideas proposed in this chapter are closely related to the work reported in (Ke and Lam, 2016), where the authors design binary matrices by solving an optimization problem that directly minimizes the error between the PCA matrix and its binary version using the Frobenius norm. However, (Ke and Lam, 2016) focused on data captured by $L^3$-imaging and solve the Frobenius-based optimization problem by using the *sign* operator to force the binary restriction. In contrast, this work focuses on the correct acquisition of compressed spectral images through real implementable optical architectures and casts the Frobenius-based optimization problem as a non-convex optimization problem that maximizes the variance explained by the binary principal components. To solve it, a greedy-search-based algorithm is proposed.

**2.1.2. Contributions.** The main contribution of this chapter is the development of a methodology to design binary sensing matrices, suitable for CSI architectures, using the structure-

preserving PCA properties from the covariance matrix. The proposed methodology includes 3 steps: first, a set of conventional random measurements/projections are captured in order to estimate the covariance matrix $\mathbf{\Sigma}$ of the spectral signal, if it is unknown; second, the subsequent sensing matrix is designed by solving a non-convex optimization problem that maximizes the variance explained by the approximated binary principal component matrix; finally, new compressive measurements are acquired with the designed matrix. Subsequently, the underlying data cube is reconstructed using the concatenation of both kind of compressive measurements i.e. those acquired with random and designed matrix, in a single linear system. Theoretical results show that the RIP constant can be considerably small when the sensing matrix is related to the eigenvectors. Additionally, computational results show an improvement in the reconstruction quality of up to 3 dB in terms of PSNR. This chapter is organized as follows: Section 2.2 poses the non-convex optimization problem to find the binary principal components. Section 2.3 shows the theoretical results, specifically the RIP analysis for the proposed sensing matrix. Finally, Section 2.4 shows the simulated experiments, and Section 2.5 discusses the findings, before concluding this work in Section 2.6.

## 2.2. Sensing matrix design via binary PCA

Let $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_n]$ be a matrix representation of the spectral image, where $\mathbf{f}_i \in \mathbb{R}^l$ is the $i^{th}$ spectral pixel of the image with $l$ bands. PCA projects the data into a subspace spanned by its own eigenvectors, such that a pixel $\mathbf{f}_i$ is projected as $\tilde{\mathbf{f}}_i = \mathbf{W}^T \mathbf{f}_i$, where the columns of $\mathbf{W}$ are the eigenvectors associated to the covariance matrix $\mathbf{\Sigma} = \mathbf{F}\mathbf{F}^T/n$, assuming that $\mathbf{F}$ is zero mean. The dimensionality reduction is achieved by constructing a matrix $\mathbf{W}_m \in \mathbb{R}^{l \times m}$ that contains only the $m$

eigenvectors associated with the $m$ largest eigenvalues of $\mathbf{\Sigma}$. Therefore, the projection of the matrix $\mathbf{F}$ can be obtained as

$$\tilde{\mathbf{F}} = \mathbf{W}_m^T \mathbf{F}. \tag{30}$$

PCA plays an important role in data dimensionality reduction and compression. The main advantage of PCA is that it preserves the structure of the data in a lower dimension by minimizing the error given by the signal $\mathbf{F}$ and its orthonormal projection in the low-dimensional subspace $\mathbf{W}_m \mathbf{W}_m^T \mathbf{F}$. However, PCA is data dependent, thus, it requires the encoder to capture and calculate the projection matrix before the compression procedure can be applied. On the other hand, CSI is a framework used to capture and compress data directly in the detector, allowing to accurately recover them by assuming some specific behavior as sparsity, low-rank or eigenvalues eccentricity (Arce et al., 2014b; Candes and Wakin, 2008; Li and Fowler, 2011; Bioucas-Dias et al., 2014a). Let $\mathbf{Q} \in \mathbb{R}^{l \times m}$ be a sampling matrix with $m < l$, thus, the noiseless sensing problem in CSI can be modeled as

$$\mathbf{Y} = \mathbf{Q}^T \mathbf{F}, \tag{31}$$

where $\mathbf{Y} \in \mathbb{R}^{m \times n}$ are the compressed measurements. Note that CSI does not require any prior knowledge of the data, and data compression is achieved without any calculation performed in the detector side. Note however that, the matrix $\mathbf{Q}$ must be binary since implementable CSI architectures use optical devices that modulate the input source with binary patterns. A traditional

optimization problem to recover the signal $\mathbf{F}$ from $\mathbf{Y}$ is given by

$$\mathbf{F} = \underset{\mathbf{F}}{\operatorname{argmin}} \quad ||\mathbf{Y} - \mathbf{Q}^T\mathbf{F}||_F^2 + \tau||\mathbf{\Psi}\text{vec}(\mathbf{F})||_1, \tag{32}$$

where $||\cdot||_F$ and $||\cdot||_1$ represent the Frobenius and $\ell_1$ -norms respectively, $\mathbf{\Psi}$ is a sparsity-promoting representation basis, and $\text{vec}(\cdot)$ represents the vectorization of a matrix.

Paraphrasing, the objective of this work is to design the matrix $\mathbf{Q}$ in (31), so that, it approximately behaves as $\mathbf{W}_m$ in (30). The motivation of making $\mathbf{Q} \approx \mathbf{W}_m$ is to preserve the $\ell_2$-norm of the data in the low dimensional space, which translates to a better RIP constant in the compressive sensing sense and thus to a better quality of the image reconstruction (as will be discussed in Section 2.3.1). The matrix $\mathbf{W}_m$ is conventionally found by solving the optimization problem (Kwak, 2008)

$$\begin{aligned} \mathbf{W}_m = \underset{\mathbf{W}_m}{\text{argmin}} \quad & ||\mathbf{F} - \mathbf{W}_m\mathbf{W}_m^T\mathbf{F}||_F^2 \\ \text{subject to} \quad & \mathbf{W}_m^T\mathbf{W}_m = \mathbf{I}, \end{aligned} \tag{33}$$

where $\mathbf{I} \in \mathbb{R}^{m\times m}$ is an identity matrix. This problem can be easily solved with an alternate gradient algorithm, however, a binary restriction must be added to meet the requirements of the CSI architectures. To find the binary matrix, we first solve the intermediate problem, obtained by adding the restriction $Q_{k,j} \in \{0, 1/\sqrt{b_j}\}$ to (33), that limits the entries of the columns of the matrix. This

leads to the problem

$$\begin{aligned} \mathbf{Q}_n = \underset{\mathbf{Q}_n, b_j}{\arg\min} \quad & ||\mathbf{F} - \mathbf{Q}_n \mathbf{Q}_n^T \mathbf{F}||_F^2 \\ \text{subject to} \quad & \mathbf{Q}_n^T \mathbf{Q}_n = \mathbf{I},\ Q_{k,j} \in \{0, 1/\sqrt{b_j}\}, \end{aligned} \tag{34}$$

where $\mathbf{Q}_n$ is the designed matrix, $b_j$ is the number of non-zero entries in its $j^{th}$ column, for $k = 1, 2, \cdots, l$ and $j = 1, 2, \cdots, \tilde{m}$, with $\tilde{m}$ the number of binary vectors. Note that, although the solution of (34) is not binary, in the sense that the entries of $\mathbf{Q}_n$ can take more than two values, it is indeed a binary matrix with normalized columns. The binary matrix can be obtained as $\mathbf{Q} = [\sqrt{b}_1 \mathbf{q}_1, \cdots, \sqrt{b}_{\tilde{m}} \mathbf{q}_{\tilde{m}}]$, since $\mathbf{Q}$ spans the same subspace of $\mathbf{Q}_n$. Solving (34) is hard due to the non-convexity entailed by the binary restriction. Therefore, we propose to solve a maximization-based problem that allows to directly design each binary vector, without relying on threshold operators. First, note that the problem in (33) is equivalent to (Jiang et al., 2015; Journée et al., 2008)

$$\begin{aligned} \mathbf{w}_j = \underset{\mathbf{w}_j}{\arg\max} \quad & \mathbf{w}_j^T \boldsymbol{\Sigma} \mathbf{w}_j \\ \text{subject to} \quad & ||\mathbf{w}_j||_2^2 = 1, \end{aligned} \tag{35}$$

where $\mathbf{w}_j \in \mathbb{R}^l$ is the $j^{th}$ column of the matrix $\mathbf{W}_m$. Problem (35) estimates a single eigenvector at a time, and then deflates the covariance matrix with the expression $\boldsymbol{\Sigma}_j = \boldsymbol{\Sigma}_{j-1} - \mathbf{w}_j \mathbf{w}_j^T \boldsymbol{\Sigma}_{j-1} \mathbf{w}_j^T \mathbf{w}_j$, to remove the influence of the already estimated eigenvector. This is the procedure used in the power iteration method. Then, by adding the binary constraint, problem (35) becomes

$$\begin{aligned} \mathbf{q}_j = \underset{\mathbf{q}_j, b_j}{\text{argmax}} \quad & \mathbf{q}_j^T \boldsymbol{\Sigma} \mathbf{q}_j \\ \text{subject to} \quad & q_j^k \in \{0, 1/\sqrt{b_j}\}, \end{aligned} \tag{36}$$

where $q_j^k$ is the $k^{th}$ entry of the $j^{th}$ column of $\mathbf{Q}_n$. The problem in (36) aims to estimate the subspace spanned by the binary vectors $\mathbf{q}_j$, that maximize the variance of the data concentrated in $\boldsymbol{\Sigma}$. Therefore, $\mathbf{q}_j$ is regarded as a binary principal component. Note that the covariance matrix $\boldsymbol{\Sigma}$ is unknown, since it depends on the data. Thus, a set of random projections must be acquired at first in order to estimate it. For this, there exist multiple algorithms in the literature that can be used (Fowler, 2009; Qi and Hughes, 2012; Bioucas-Dias et al., 2014a). This work employs the CPPCA approach, proposed in (Fowler, 2009), due to its speed and reliability in estimating the covariance matrix. This approach will be addressed in detail in Section 2.3.2.

Additionally, note that (36) is non-convex because the convex combination of two binary vectors does not necessarily result in a binary vector, i.e. $\lambda \mathbf{q}_j + (1-\lambda)\mathbf{q}_{j'}$ is not necessarily binary. Thus, Algorithm 2.1 has been developed to approximate the solution of (36). This algorithm is greedy-search-based since it iteratively looks for the best position within the vector that maximizes the objective function. In spite of its greedy characteristics, the proposed algorithm exhibits good performance, as it will be shown in the simulations, and furthermore, it is parameter-free.

In detail, Algorithm 2.1 requires an initial estimation of the covariance matrix $\boldsymbol{\Sigma}_1$ and the number of binary vectors $\tilde{m}$ to be designed. Then, it initializes the coding pattern $\mathbf{q}_j$ with a vector of zeros (Line 3). In each iteration, it looks for the best position where a binary value of $1/\sqrt{b}$

maximizes the objective function, but keeping fixed the positions previously found. That is, in each iteration, a previously selected position is never evaluated or tested again. The for-loop in Line 4 only iterates over $(l/\tilde{m})$ to promote the resulting vectors to be orthonormal to each other, which implies that the transmittance of the resulting binary vector is forced to be less or equal than $1/\tilde{m}$. Note that, the transmittance is defined as the ratio between the number of nonzero elements and the total number of elements. In Line 6, a value of one is placed in a certain position and the objective function is calculated (Line 7) and compared with its previous value. If the current position maximizes the function, it is considered as a candidate (Line 8 to 10). To test other positions, the one is removed in Line 11. At the end of the loop, the best position is stored in the

**Algorithm 2.1** B-PCA: Binary PCA estimation

1: **input:** $\mathbf{\Sigma}_1, \tilde{m}$
2: **for** $j = 1$ to $\tilde{m}$ **do**
3: $\quad \mathbf{p}_j \leftarrow \mathbf{0}$; $max \leftarrow 0$; $list = 1, 2, \cdots, l$
4: $\quad$ **for** $k = 1$ to round$(l/\tilde{m})$ **do** *impose transmit.* $1/\tilde{m}$
5: $\quad\quad$ **for** $i =$ each element in *list* **do**
6: $\quad\quad\quad \mathbf{p}_j^i \leftarrow 1$ *place a one in the* $i^{th}$ *position*
7: $\quad\quad\quad c^k \leftarrow \frac{\mathbf{p}_j^T}{||\mathbf{p}_j||} \mathbf{\Sigma}_j \frac{\mathbf{p}_j}{||\mathbf{p}_j||}$ *objective function* (36)
8: $\quad\quad\quad$ **if** $c^k > max$ **then**
9: $\quad\quad\quad\quad max = c^k$; $index = i$
10: $\quad\quad\quad$ **end if**
11: $\quad\quad\quad \mathbf{p}_j^i \leftarrow 0$ *remove the one value*
12: $\quad\quad$ **end for**
13: $\quad\quad \mathbf{p}_j^{index} \leftarrow 1$ *place the one in the best position*
14: $\quad\quad list.remove(index)$; $index \leftarrow 0$
15: $\quad$ **end for**
16: $\quad \mathbf{Q} \leftarrow \frac{\mathbf{q}_j \mathbf{q}_j^T}{||\mathbf{q}_j||^2}$
17: $\quad \mathbf{\Sigma}_{j+1} \leftarrow \mathbf{\Sigma}_j - \mathbf{\Sigma}_j \mathbf{Q} - \mathbf{Q}\mathbf{\Sigma}_j + \mathbf{Q}\mathbf{\Sigma}_j \mathbf{Q}$
18: **end for**
19: **output:** $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{\tilde{m}}]$

variable *index*, such that a value of one is fixed there (Line 13). After that, the covariance matrix $\mathbf{\Sigma}$ is deflated by subtracting the influence of the vector $\mathbf{q}_j$ (Line 16 and 17) and the for-loop continues. The proof that expression in Line 17 subtracts the influence of the subspace spanned by $\mathbf{q}$ is shown in Appendix 6, in the supplementary material. Algorithm 2.1 has computational complexity $O(l^3)$ and its proof is deferred to Appendix 6, in the supplementary material. Note that, although the complexity of the algorithm is dominated by the cubic term, spectral images usually span along few hundreds of bands, thus it does not considerably increase the computational complexity of the problem. Additionally, an analysis of the convergence of the algorithm is presented in Appendix 6, in the supplementary material, explaining why the restriction $q_j^k \in \{0, 1/\sqrt{b_j}\}$ is needed. Furthermore, a flowchart that describes the Algorithm 2.1 is shown in Fig. 45, in the supplementary material, and a Matlab code implementation of this algorithm can be found and tested at `https://codeocean.com/capsule/8658864/`.

### 2.3. Theoretical Results

**2.3.1. Restricted Isometry Property Analysis.** This section shows that the use of PCA satisfies the RIP in specific cases. It is important to show that the RIP holds when the sensing matrix $\mathbf{Q} = \mathbf{W}_m$, and the signal can be accurately represented in the subspace given by span$(\mathbf{W}_m)$, linking the use of PCA to the CSI sensing procedure. Additionally, it sets a relationship between the variance and the expected value of the pixel norm linking PCA and the RIP.

**Theorem 1.** Let $\mathbf{f} = \mathbf{W}\boldsymbol{\theta} \in \mathbb{R}^l$ be a pixel of a spectral image and $\mathbf{W} \in \mathbb{R}^{l \times l}$ be the matrix whose

columns are the eigenvectors of the covariance matrix $\mathbf{\Sigma} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T$. It holds that

$$(1-\delta_m)||\boldsymbol{\theta}_P||_2^2 \leq ||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2 \leq (1+\delta_m)||\boldsymbol{\theta}_P||_2^2, \tag{37}$$

with

$$\delta_m = |P_2|\lambda_{m+1}/(\sum_{i\in P_1}\lambda_i + \sum_{i\in P_2}\lambda_i), \tag{38}$$

and $P_1 \subset \{1,2,\cdots,m\}, P_2 \subset \{m+1,\cdots,l\}$, $\mathbf{A}_P$ is a sub-matrix whose columns are a subset of the columns of $\mathbf{A} = \mathbf{W}_m^T\mathbf{W}$ , $P = P_1 \cup P_2$, and $\lambda_i$ is the $i^{th}$ eigenvalue of the covariance matrix of the data. Note that, in most natural scenes, pixels can be accurately represented in a low dimensional space, hence, most of the information is kept in the first eigenvalues which implies that $0 < \delta_m \ll 1$.

The proof of this theorem is deferred to Appendix 6, in the supplementary material.

Theorem 1 shows that, when the eigenvectors are used as linear projectors, the norm of a vector changes in proportion to the least important eigenvalues. Further, note that binary versions of the eigenvectors are used, given by $\mathbf{Q}$ and thus, (37) does not hold since $\mathbf{Q}^T\mathbf{W} \neq \mathbf{I}$. The RIP for $\mathbf{Q}$ is stated in the Corollary 1.1

**Corollary 1.1.** For an arbitrary sensing matrix $\mathbf{A}$, the RIP constant is bounded by $\delta_{\tilde{m}} = (\sum_{k\in P_1}\lambda_k|\beta_k| + \sum_{k'\in P_2}\lambda_{k'}|\beta_{k'}|)/(\sum_{k\in P_1}\lambda_k + \sum_{k'\in P_2}\lambda_{k'})$ with $|\beta_k| = |1-\sum_{i=1}^{\tilde{m}}(a_k^i)^2| = |1-\sum_{i=1}^{\tilde{m}}(\mathbf{q}_k^T\mathbf{w}_i)^2|$ and $|\beta_{k'}| = |1-\sum_{i=1}^{\tilde{m}}(a_{k'}^i)^2| = |1-\sum_{i=1}^{\tilde{m}}(\mathbf{q}_{k'}^T\mathbf{w}_i)^2|$ for $k = 1,\cdots,\tilde{m}$ and $k = \tilde{m}+1,\cdots,l$

The proof of the corollary is detailed in Appendix 6, in the supplementary material.

Corollary 1.1 shows that if the columns of the sensing matrix are unit-norm, the RIP con-

stant is small. However, for the case of the eigenvectors, it is enough to ensure that this holds for the columns related to the largest eigenvalues. Thus, in order to guarantee that $\delta_{\tilde{m}}$ is small, $|\beta| = |1 - \sum_{i=1}^{\tilde{m}} (a_k^i)^2| = 0$ should hold, i.e. $\sum_{i=1}^{\tilde{m}} (a_k^i)^2 = \sum_{i=1}^{\tilde{m}} (\mathbf{q}_k^T \mathbf{w}_i)^2 = 1$. Intuitively, it can be seen that if $\mathbf{q}_k \approx \mathbf{w}_k$, $a_k^k \approx 1$ and

$$\sum_{i=1}^{\tilde{m}} (a_k^i)^2 \approx \sum_{i=1}^{k-1} (a_k^i)^2 + 1 + \sum_{i=k+1}^{\tilde{m}} (a_k^i)^2. \tag{39}$$

Therefore, if $\mathbf{q}_i$ is orthogonal to $\mathbf{q}_k$, for $i \neq k$, we get that

$$\sum_{i=1}^{\tilde{m}} (a_k^i)^2 \to 1, \text{ and thus, } \delta_{\tilde{m}} \to 0. \tag{40}$$

**2.3.2. Covariance Matrix Estimation.** For the covariance matrix estimation, the CPPCA approach was adopted (Fowler, 2009). For that, define

$$\mathbf{F}_i = [\mathbf{f}_{\Omega_i^1}, \cdots, \mathbf{f}_{\Omega_{i'}^{n/p}}], \Omega_i^j \neq \Omega_k^l, i \neq k, \forall_{i,k}, \tag{41}$$

as a subset of the hyperspectral image $\mathbf{F}$ introduced in (30), where $\Omega_i \subset \Omega = \{1, \ldots, n\}$, and $\Omega_i^j$ refers to the $j^{th}$ element in the subset $\Omega_i$ with $i = 1, \ldots, p$ such that $p \ll n$. Additionally, not only one projection matrix $\mathbf{Q} \in \mathbb{R}^{l \times v}$ is used, but $p$ different projection matrices $\mathbf{Q}_i \in \mathbb{R}^{l \times v}$, which allow to rewrite the problem in (31) as

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{H}}\tilde{\mathbf{F}}, \tag{42}$$

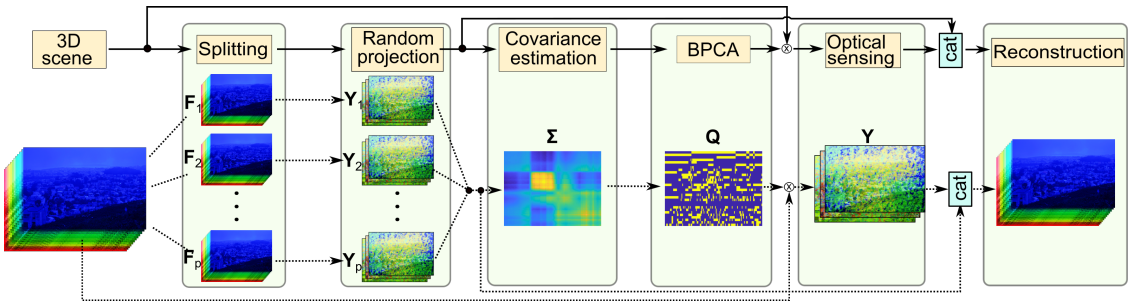


*Figure 7.* Flowchart for the sensing and reconstruction algorithm. Solid lines show a block diagram of the procedure and dotted lines show graphically the same procedure. First, the image is divided into $p$ subsets and projected using random matrices. The covariance matrix is estimated using these random projections and it is used to design the binary sensing matrix.

where $\tilde{\mathbf{Y}} = [\mathbf{Y}_1^T, \cdots, \mathbf{Y}_p^T]^T$, $\mathbf{Y}_i \in \mathbb{R}^{v \times n/p}$, $\tilde{\mathbf{H}} = \text{diag}(\mathbf{Q}_1^T, \cdots, \mathbf{Q}_p^T)$, $\tilde{\mathbf{F}} = [\mathbf{F}_1^T, \cdots, \mathbf{F}_p^T]$ and $v$ is the number of random acquisitions or randomly generated rows. This $p$-partition sensing approach is borrowed from (Martín et al., 2015) following the sensing strategy of HYCA (hyperspectral coded aperture).

This sensing strategy can be implemented in CSI optical architectures, such as, the 3D-CASSI (Cao et al., 2016) and the DD-CASSI (Gehm et al., 2007a). Using this approach, the covariance matrix can be rapidly estimated from a set of random projections using a projection-onto-convex-sets based algorithm (POCS). The reader is encouraged to check (Fowler, 2009) for more details on POCS. Additionally, the data must be centered, but since the spectral images take values between 0 and $2^{bits} - 1$, the zero mean assumption is unrealistic. Reconstructing the image in order to estimate its mean will result in high computational costs, therefore, the mean should be calculated directly from the compressive measurements using the randomly generated sensing matrices as [8]

$$\widehat{\mathbf{f}} = \alpha \sum_{i=1}^{p} \sum_{j=1}^{k} \mathbf{Q}_i^T (\mathbf{Q}_i \mathbf{Q}_i^T)^{-1} \mathbf{y}_i^j, \tag{43}$$

where $\alpha = m/n$, $\mathbf{y}_i^j$ is the *j-th* pixel in the *i-th* partition or subset, and $k$ is the number of pixels in each partition $p$, such that $pk = n$. It was proved that Eq. (43) converges to the true mean when $n \to \infty$ (Qi and Hughes, 2012). Once the mean is estimated, the measurements are centered by subtracting it as $\tilde{\mathbf{Y}}_i = \mathbf{Y}_i - \mathbf{Q}_i^T(\widehat{\mathbf{f}} \otimes \mathbf{1}^T)$, where $\otimes$ represents the Kronecker product, and $\mathbf{1} \in \mathbb{R}^k$ is a k-long one-valued vector. This operation replicates the mean and subtracts it from the compressive measurements. Performing this operation only once usually does not produce very accurate results, however, estimating the mean and subtracting it multiple times in a for-loop usually produces better results (Qi and Hughes, 2012).

**2.3.3. Sensing and Reconstruction Methodology.** In summary, the proposed sensing and reconstruction methodology is detailed in Algorithm 2.2 and sketched in Fig. 7. First, $p$ randomly generated matrices $\{\mathbf{Q}_i\}_{i=1}^p \in \mathbb{R}^{l \times v}$, following a Bernoulli distribution, are used as the initial sensing matrices. Using the random compressed measurements $\{\mathbf{Y}_i\}_{i=1}^p$ of each disjoint subset $\{\mathbf{F}_i\}_{i=1}^p$, the covariance matrix is estimated (Lines 6-7) and the subsequent matrix is designed following Algorithm 1 (Line 8). Afterwards, the designed matrix $\mathbf{Q} \in \mathbb{R}^{l \times \tilde{m}}$ is concatenated with the initial random matrices $\mathbf{Q}_i$, as stated in Lines 9 to 12, and the sensing process is repeated using the resulting matrix in Line 13. The concatenation of random and designed matrices is performed in order to improve the condition of the problem. This is, the rank of the designed matrix is at most $\tilde{m}$ and the rank of the concatenation is at most $v + \tilde{m}$. However, if the covariance matrix is known a-priori, the random measurements are not required. Finally, the optimization problem to recover the image is solved in Line 14. The latter can be done using algorithms like GPSR (Figueiredo et al., 2007), SALSA (Afonso et al., 2010), or SpaRSA (Wright et al., 2009).

**Algorithm 2.2** Proposed sensing and reconstruction protocol

1: **input:** $\{\mathbf{F}_i\}_{i=1}^{p}$, $\tilde{m}$
2: **for** $i = 1$ to $p$ **do**
3: $\quad \mathbf{Q}_i \leftarrow randbinary(l, v)$
4: $\quad \mathbf{Y}_i \leftarrow \mathbf{Q}_i^T \mathbf{F}_i$
5: **end for**
6: $\{\tilde{\mathbf{F}}_i\}_1^p = \text{CPPCA}(\{\mathbf{Q}_i\}_1^p, \{\mathbf{Y}_i\}_1^p)$ ▷ *Image estimation*
7: $\tilde{\mathbf{\Sigma}} = \frac{1}{n}\sum_{i=1}^{p} \tilde{\mathbf{F}}_i \tilde{\mathbf{F}}_i^T$ ▷ *Estimate covariance matrix*
8: $\mathbf{Q} \leftarrow \text{B-PCA}(\tilde{\mathbf{\Sigma}}, \tilde{m})$ ▷ *Find binary principal components*
9: **for** $i = 1$ to $p$ **do**
10: $\quad \overline{\mathbf{Q}}_i \leftarrow [\mathbf{Q}_i^T, \mathbf{Q}^T]^T$ ▷ *Concatenate sensing matrices*
11: $\quad \mathbf{Y}_i \leftarrow [\mathbf{Y}_i^T, (\mathbf{Q}^T \mathbf{F}_i)^T]^T$ ▷ *Update measurements*
12: **end for**
13: $\tilde{\mathbf{H}} \leftarrow \text{diag}(\overline{\mathbf{Q}}_1^T, \cdots \overline{\mathbf{Q}}_p^T)$
14: $\tilde{\mathbf{F}} \leftarrow \text{argmin}_{\tilde{\mathbf{F}}} ||\tilde{\mathbf{Y}} - \tilde{\mathbf{H}}\tilde{\mathbf{F}}||_F^2 + \tau ||\mathbf{\Psi}\text{vec}(\tilde{\mathbf{F}})||_1$
15: **Output:** $\tilde{\mathbf{F}}$

### 2.4. Simulations and Results

In this section, two hyperspectral images are used as input, to demonstrate the effectiveness of the designed binary PCA matrices. The hyperspectral scenes are the Urban dataset (Zhu et al., 2014) with $256 \times 256$ pixels of spatial resolution and $l = 128$ spectral bands, and a section of the Pavia centre recorded by the ROSIS sensor (Mueller et al., 2002), with $512 \times 512$ pixels of spatial resolution and $l = 102$ spectral bands. A single spectral band and three pixels of each dataset are shown in Figs. 8 and 9, respectively.

The random sensing matrices $\mathbf{Q}_i$ are generated following a Bernoulli distribution. The reconstruction of the full datacube from the set of random+designed measurements, Line 14 in Algorithm 2, is performed using the Split Augmented Lagrangian Shrinkage Algorithm (SALSA) (Afonso et al., 2010), using a signal sparsity prior over the 3D Kronecker basis formed by the

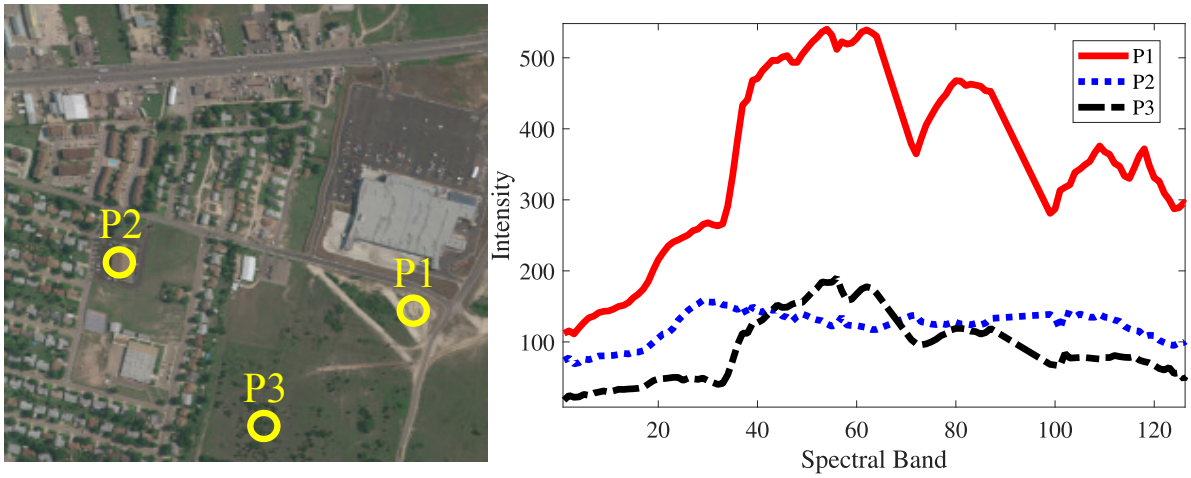


*Figure 8.* Urban dataset. (Left) RGB composite of Urban datset. (Right) Three spectral signatures at different pixels of the image.

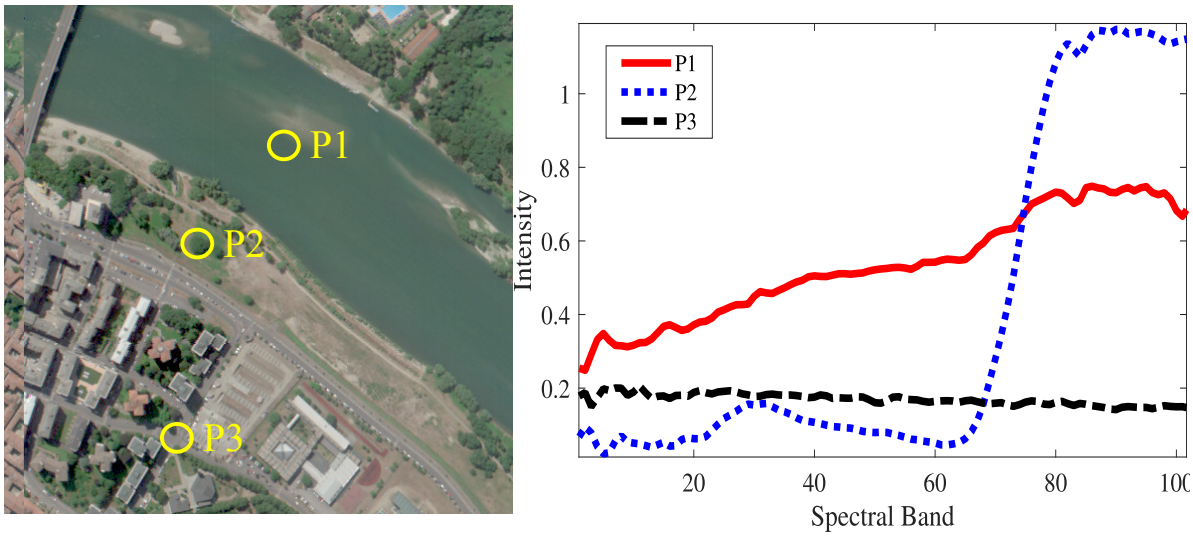


*Figure 9.* Pavia centre dataset. (Left) RGB composite of Pavia datset. (Right) Three spectral signatures at different pixels of the image.

Wavelet 2D Symlet 8 and the Discrete Cosine transform (DCT). The relative variation of the objective function is used as the stopping criterion, and it is set to 1e$-$5. The number of partitions is set to $p = l$, for simplicity. The performance is measured in terms of the spectral peak signal to noise ratio (PSNR), defined as PSNR $= 1/n \sum_{i=1}^{n} (10 \log A^2 / ||\mathbf{f}_i - \tilde{\mathbf{f}}_i||_2^2)$, where $A$ is the maximum

amplitude value of a pixel, $\mathbf{f}_i$ is the $i^{th}$ pixel of the reference image and $\tilde{\mathbf{f}}_i$ is the reconstructed pixel, the mean squared error (MSE), defined as $1/n\sum_{i=1}^{n}||\mathbf{f}_i-\tilde{\mathbf{f}}_i||_2^2$, and the spectral angle mapper (SAM) defined as $\arccos(\mathbf{f}^T\tilde{\mathbf{f}}/(||\mathbf{f}||||\tilde{\mathbf{f}}||))(180/\pi)$.

**2.4.1. Mean estimation.** As hyperspectral images do not have zero mean, it is first estimated following Eq. (43), and then subtracted from the measurements. For Urban and Pavia, we test the quality of this estimation by setting $p = l$ and varying the number of random projections from 6 to 18. The normalized mean squared error, $NMSE = MSE/||\mathbf{f}||$, is used to measure the quality of the estimated means. Table 1 shows the overall results in terms of the NMSE and SAM, where it can be seen that the estimated mean does not differ much from the true mean as the number of projections increases.

Table 1
*NMSE of the estimated mean varying the number of shots*

| | | **Shots** | | | |
|---|---|---|---|---|---|
| **Image** | **Metric** | 6 | 10 | 14 | 18 |
| **Urban** | NMSE | 0.182 | 0.116 | 0.074 | 0.046 |
| | SAM | 10.402 | 6.470 | 4.172 | 2.574 |
| **Pavia** | NMSE | 0.171 | 0.095 | 0.053 | 0.031 |
| | SAM | 9.3690 | 5.1879 | 3.0096 | 1.6407 |

**2.4.2. Quality of the PCA-based Designed Matrices.** First, the quality of the designed sensing matrices is evaluated by testing only the $\ell_2$ approximation in (32), setting $\tau = 0$. Note that the reconstruction using only the $\ell_2$-term can be done in closed-form via the Moore-Penrose pseudo-inverse. The results attained with the designed matrices are compared against the ones with randomly generated matrices, the ones generated with the algorithm proposed in (Ke and

Table 2
*Overall performance of the sensing matrices. The column "Random" represents the randomly generated matrices, "Estimated $\mathbf{\Sigma}$" represents the designed matrices using the estimated covariance matrix and "A-priori $\mathbf{\Sigma}$" the designed matrices but using the true covariance matrix.*

| Image | Metric | Random | Estimated $\mathbf{\Sigma}$ | | A-priori $\mathbf{\Sigma}$ | |
|---|---|---|---|---|---|---|
| | | | Prop. | Qpca | Prop. | Qpca |
| **Urban** | MSE | 292.03 | 106.1 | 240.6 | 105.3 | 230.7 |
| | PSNR | 10.31 | 19.27 | 11.51 | 20.31 | 12.82 |
| | SAM | 35.86 | 4.692 | 27.50 | 4.59 | 21.64 |
| **Pavia** | MSE | 324.7 | 147.9 | 212.1 | 135.2 | 186.3 |
| | PSNR | 10.12 | 16.30 | 11.74 | 16.47 | 13.38 |
| | SAM | 37.34 | 6.23 | 29.99 | 5.82 | 14.44 |

Lam, 2016)(which will be termed Qpca) and the ground-truth data. In this test, the results attained with the designed matrix but calculated from the ground-truth covariance matrix are also included, in order to check the induced error when it is designed directly from the random projections. Table 2 shows the overall results in terms of MSE, PSNR and SAM, and Fig. 10 depicts the behavior of the reconstruction at a specific pixel of the datasets. It can be noticed that the proposed sensing matrices achieve a better performance in terms of the three metrics, disregarding how $\mathbf{\Sigma}$ is estimated. In terms of MSE, the proposed matrices achieve a 3-fold improvement compared with random measurements, and around 2-fold in terms of PSNR. Remark that the results with the estimated $\mathbf{\Sigma}$ closely approximate the ones from the true $\mathbf{\Sigma}$, confirming the good quality of the statistics estimated from the random projections.

Additionally, the optimization problem proposed in (36) was compared with the standard PCA problem given in (35). For that, the variance explained by varying the number of vectors is

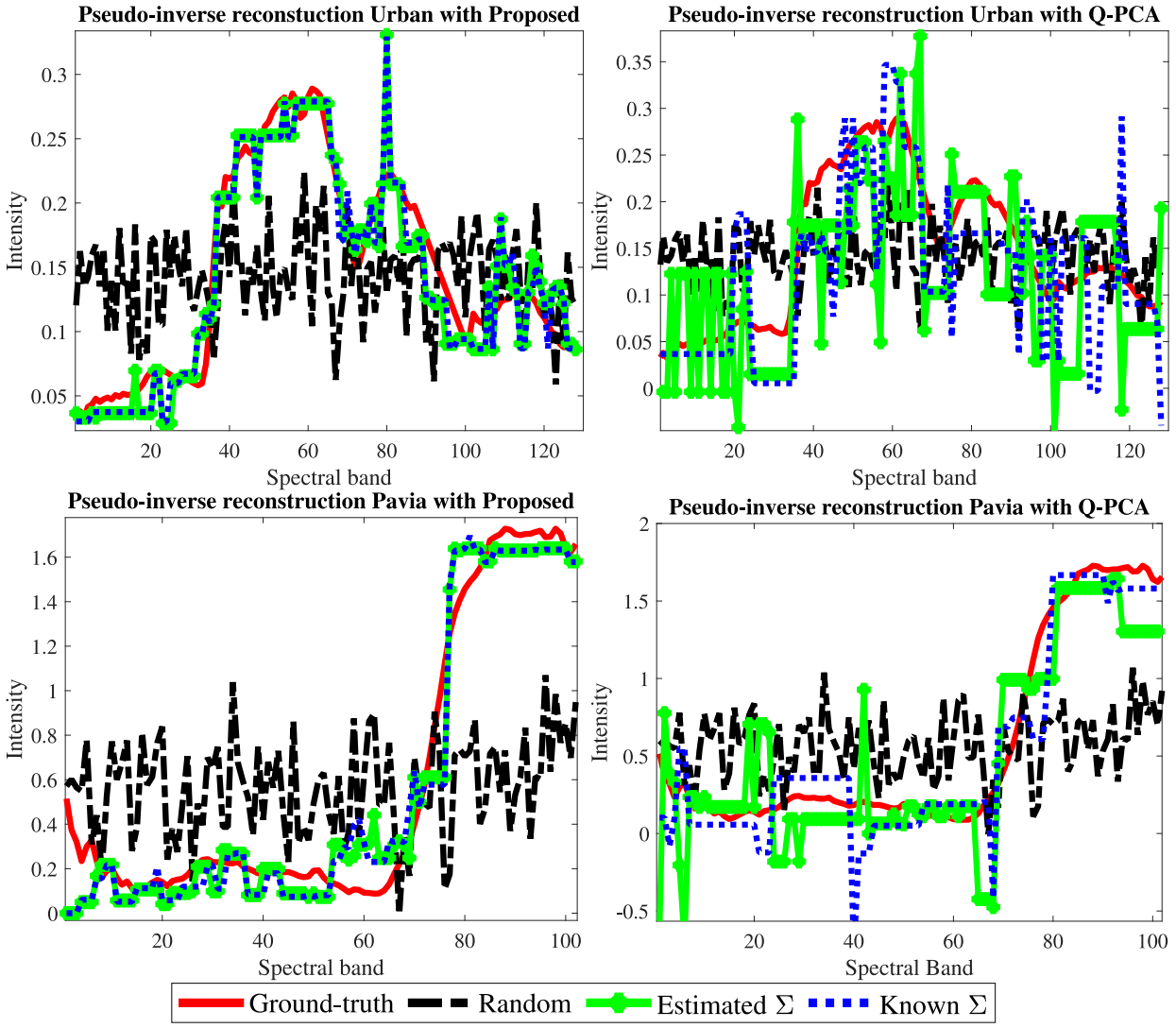


*Figure 10.* Comparison of a reconstructed pixel solving only the $\ell_2$ term in (32) via the Moore-Penrose pseudo-inverse, using 8 random measurements (black line), 8 designed measurements with prior $\mathbf{\Sigma}$ (blue lines), and designed matrix with estimated $\mathbf{\Sigma}$ (green lines). (Top-Left) Results for the Urban dataset using the proposed matrix. (Top-Right) Results for the Urban dataset using Qpca. (Bottom-left) Results for Pavia dataset using the proposed matrix. (Bottom-right) Results for Pavia dataset using Qpca.

analyzed. The variance explained or retained is defined as

$$d = \text{trace}(\mathbf{Q}^T \mathbf{\Sigma} (\mathbf{Q}^\dagger)^T) / \text{trace}(\mathbf{\Sigma}). \tag{44}$$

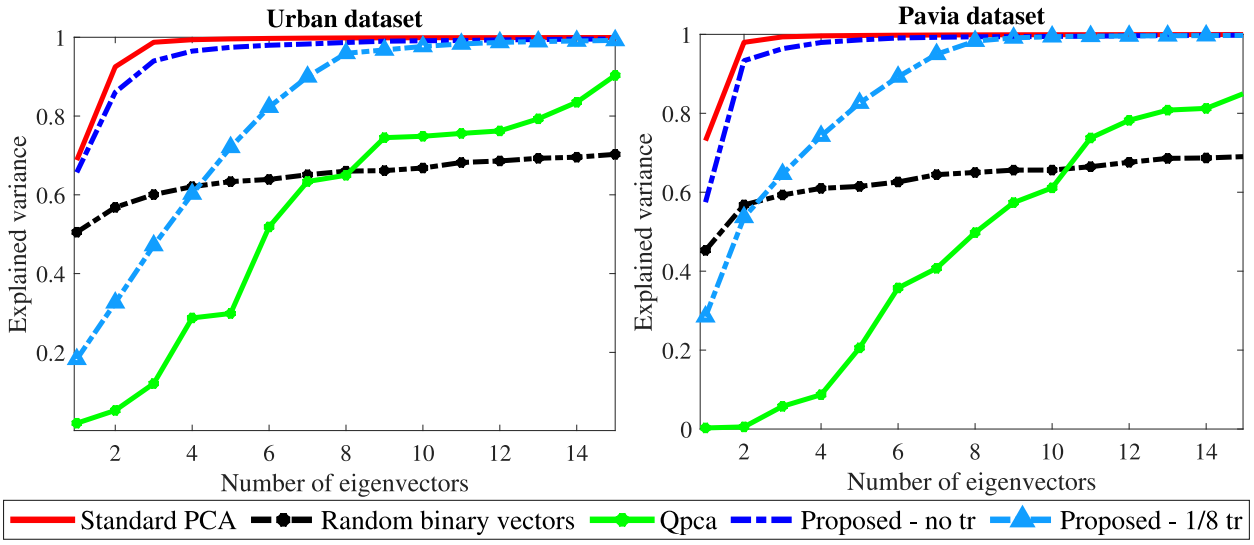


*Figure 11.* Explained variance by varying the number of eigenvectors with different methods. Red - solid line represents the variance for traditional PCA (theoretical limit). Blue dot-dashed line is used for the proposed binary eigenvectors when no transmittance restriction is imposed, dashed line with triangles represents the proposed binary eigenvectors when the transmittance is set to 12.5% (1/8), green-solid line with marks represents Qpca, whereas Black-dashed line is for a random matrix.

Figure 11 was generated by designing 35 vectors using the proposed optimization problem and the method proposed in (Ke and Lam, 2016), and then calculating the explained variance using (44) when a subset of vectors is used. Figure 11 shows that the proposed binary eigenvectors explain better the variance of the data in comparison with Qpca and random vectors when no restriction in the transmittance is imposed. For instance, for 98% of variance, standard PCA requires only 3 eigenvectors, the proposed binary PCA requires 5 to 7 depending on the image, and Qpca is not able to reach this amount of explained variance. However, as the theoretical results suggest, restricting the transmittance improves the RIP, thus the sensing matrices for the reconstruction experiments are calculated with the transmittance restriction imposed. Although the explained variance reduces, it is still larger than Qpca and random. As an example, Fig. 12 shows a realiza-

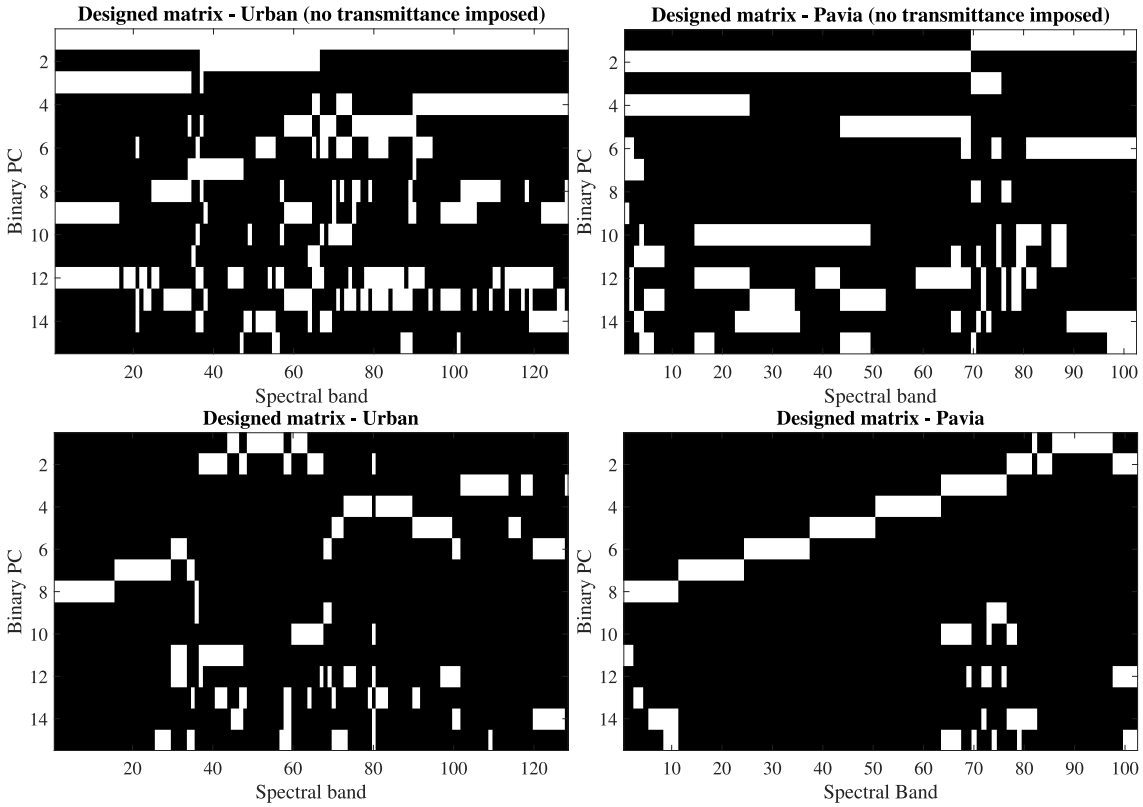


*Figure 12.* Realization of the designed matrices for Urban and Pavia datasets with 15 vectors. (Top-row) Designed matrices when no transmittance is imposed. (Bottom-row) Designed matrices with 1/8 of transmittance.

tion of the binary matrix obtained by our method, when no transmittance is imposed and when it is imposed, for both datasets.

**2.4.3. Reconstruction Performance under Noisy Scenarios.** In this experiment, the performance of the designed matrices is evaluated by reconstructing the datasets using the $\ell_2 - \ell_1$ optimization problem in (32). For this, the number of measurements is varied from 18 to 36, which roughly represents a 14% to 28% compression ratio for the Urban dataset, and a 17% to 35% compression ratio for the Pavia dataset. Intuitively, it will be ideal to use as many designed measurements as possible, however, since the covariance matrix is estimated from the

Table 3
*Relationship between the number of random and designed vectors used in the sensing procedure.*

| | | **Shots** | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Image** | **Kind** | 18 | 21 | 24 | 27 | 30 | 33 | 36 |
| **Urban** | Random | 10 | 13 | 15 | 18 | 21 | 24 | 27 |
| | Designed | 8 | 8 | 9 | 9 | 9 | 9 | 9 |
| **Pavia** | Random | 13 | 15 | 18 | 20 | 23 | 26 | 28 |
| | Designed | 5 | 6 | 6 | 7 | 7 | 7 | 8 |

compressed measurements, some of these vectors should be generated at random. For instance, when 18 measurements are being captured, one could use 9 random and 9 designed, or 10 random and 8 designed, or any other combination. However, the number of designed binary vectors $\tilde{m}$ in this work is calculated by computing the variance explained by the binary eigenvectors as in (44). Thus, by fixing this percentage to be 98%, different numbers of random and designed vectors were used for the reconstructions, as shown in Table 3.

Remark that the different proportions are due to the fact that, at some point, using more designed measurements do not contribute significantly in terms of data variance, but, as shown in Table 1, since the covariance matrix is estimated from the random measurements, increasing its number improves the estimation of $\mathbf{\Sigma}$.

Reconstruction results using the PSNR as the performance metric, for the two datasets, are summarized in Fig. 13. In this figure, two noisy scenarios are tested, with signal-to-noise-ratios (SNR) of 15 and 25 dB. Overall, the designed matrices (solid lines) outperform the random (dotted lines) by up to 3 dB in the two noisy scenarios, and up to 2 dB in comparison with Qpca (dot-dashed lines) with high compression ratios. Additionally, note that, the entries of the matrices produced

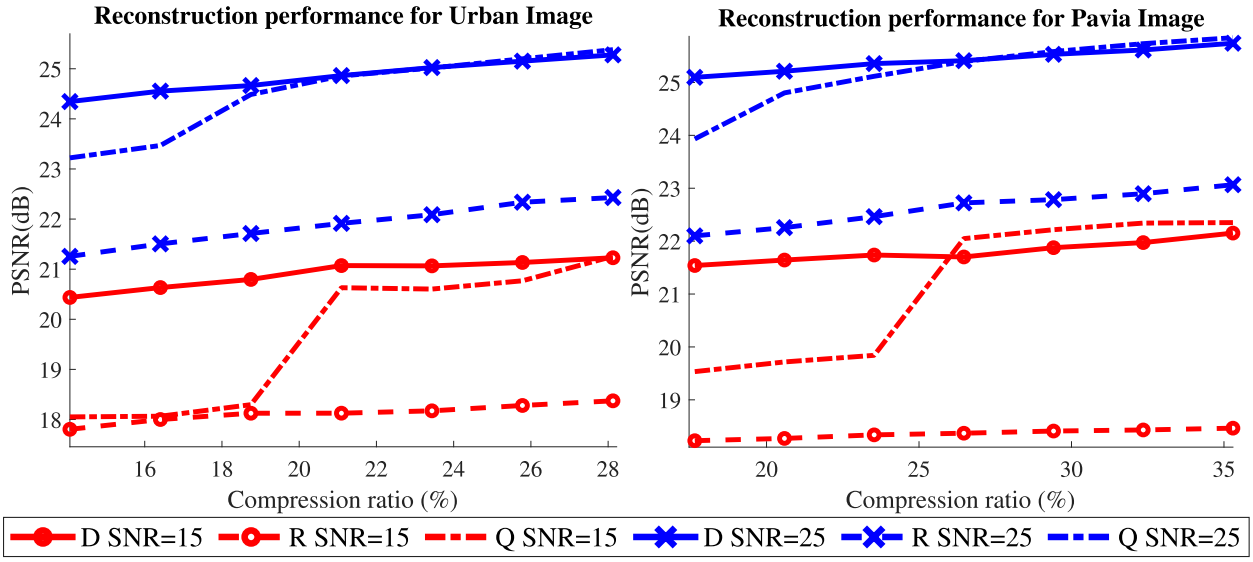


*Figure 13.* Average PSNR of the reconstructed hyperspectral datasets for various compression ratios, at 2 noise scenarios with SNR = 15 and 25 dB. “D" stands for designed (solid lines), “Q” for those proposed in (Ke and Lam, 2016) (Qpca) (dot-dashed lines) and “R" for random (dotted lines) matrices. Different colors represent different noise levels. (Left) Urban dataset. (Right) Pavia dataset

.

by (Ke and Lam, 2016) are $\{-1, 1\}$, which entail problems when being implemented, such as the requirement of an additional all-pass shot in order to produce the $-1$ code word. Furthermore, since this type of coding cannot be implemented directly, its emulation increases the noise by a factor of 5 (Pinilla et al., 2016). Under this scenario the reconstructions obtained by the Qpca in Figs. 13 to 15 would be dramatically affected.

Additionally, to visually compare the attained reconstructions, Figs. 14 and 15 show the reconstructions at two randomly selected spatial pixels, denoted as P1 and P2, and a specific spectral band when 21 measurements are captured (16% and 20% of the data for Urban and Pavia datasets, respectively). There, it can be noticed that the spectral signatures attained with the designed matrices closely resemble the ground truth, and the difference of the spectral bands show finer details

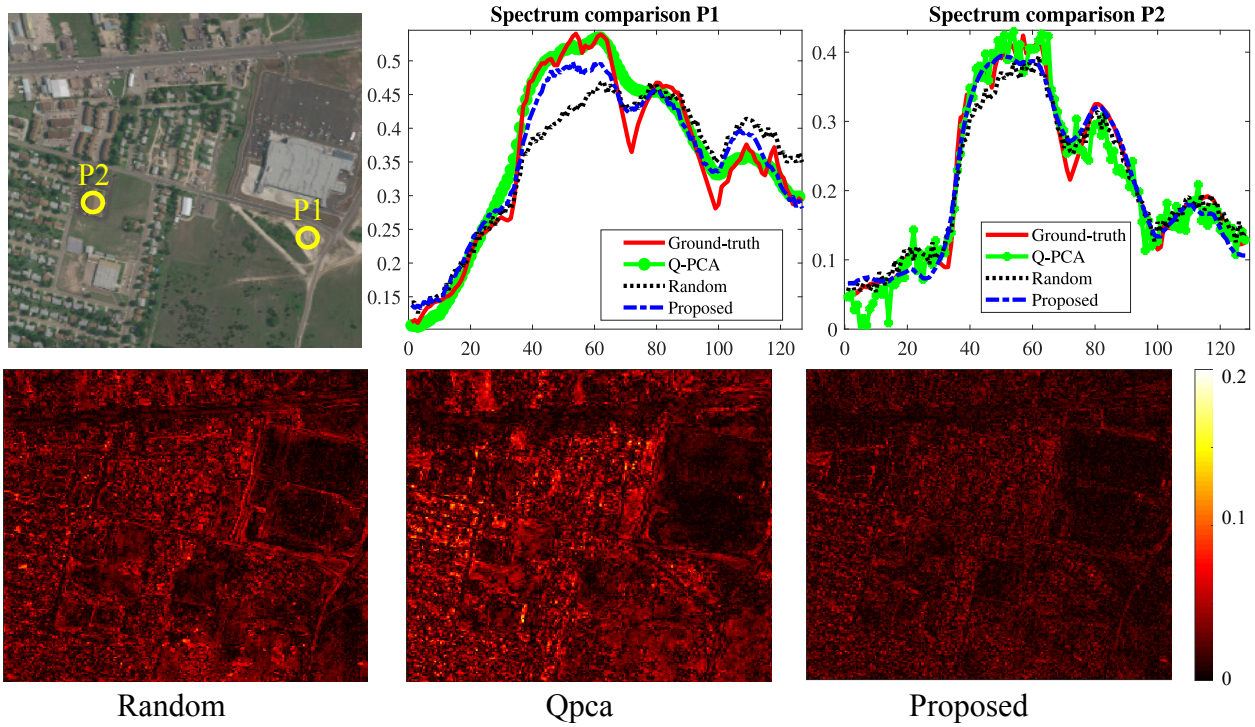


*Figure 14.* Comparison between some pixels (P1, P2) of the Urban dataset using 21 measurements (20.58% of the data, 13 random and 8 designed). (Top-left) RGB composite of the ground-truth. (Top-middle) Comparison between reconstructed and ground-truth for P1. (Top-right) Comparison between reconstructed and ground-truth for P2. (Bottom-left) Normalized residual of the reconstruction using random matrices. (Bottom-middle) Normalized residual using Qpca. (Bottom-right) Normalized residual using the proposed matrices.

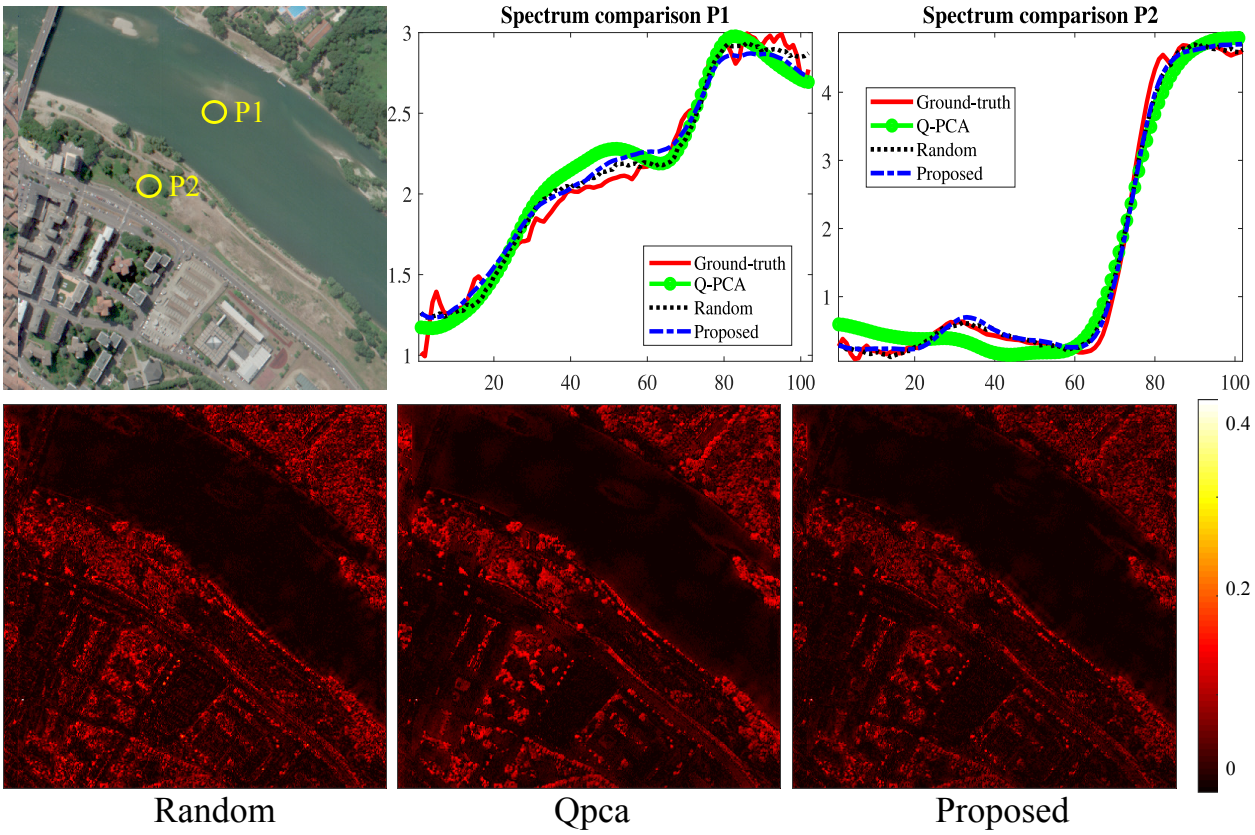


*Figure 15.* Comparison between some pixels (p1, P2) of the Pavia dataset using 21 measurements (16.41% of the data, 15 random and 6 designed). (Top-left) RGB composite of the ground-truth. (Top-middle) Comparison between reconstructed and ground-truth for P1. (Top-right) Comparison between reconstructed and ground-truth for P2. (Bottom-left) Normalized residual of the reconstruction using random matrices. (Bottom-middle) Normalized residual using Qpca. (Bottom-right) Normalized residual using the proposed matrices.

in the reconstructions attained with the designed matrices.

## 2.5. Discussion

We present a brief analysis of the differences in the explained variance of the proposed algorithm in comparison to Qpca. Additionally, it is discussed why the proposed binary eigenvectors exhibit a close behavior to those obtained with standard PCA in terms of explained variance.

**2.5.1. Explained Variance Comparison.** The proposed algorithm performs very well in terms of explained variance in contrast to Qpca. This behavior is expected since the Qpca algorithm minimizes the objective function given by

$$\begin{aligned} \arg\min_{\mathbf{D}} \quad & ||\sqrt{N}\mathbf{D}\mathbf{W}_m^T - sign(\mathbf{D}\mathbf{W}_m^T)||_F^2 \\ \text{subject to} \quad & \mathbf{D}\mathbf{D}^T = \mathbf{I}, \end{aligned} \tag{45}$$

which does not take into account the eigenvalues. Considering the eigenvalues is critical in hyperspectral imaging, since usually the first eigenvectors are much more important than the last ones. In fact, the proposed algorithm uses the covariance matrix rather than just the eigenvectors as in (Ke and Lam, 2016), which implicitly takes into account the eigenvalues. To show that, we present additional results in Fig. 16, when the matrix $\mathbf{W}_m$ is scaled by a diagonal matrix $\mathbf{B}$ such that, the first eigenvector is scaled by a larger number than the last eigenvector, i.e. $\mathbf{W}_m = \mathbf{W}_m\mathbf{B}$. Note however that by doing this, the Qpca algorithm should be adjusted since it assumes that $\mathbf{W}_m^T\mathbf{W}_m = \mathbf{I}$, and it does not hold for $\mathbf{W}_m = \mathbf{W}_m\mathbf{B}$. Nevertheless, it can be seen that by scaling the eigenvectors according to their eigenvalues, the explained variance of the resulting Qpca binary matrix improves,

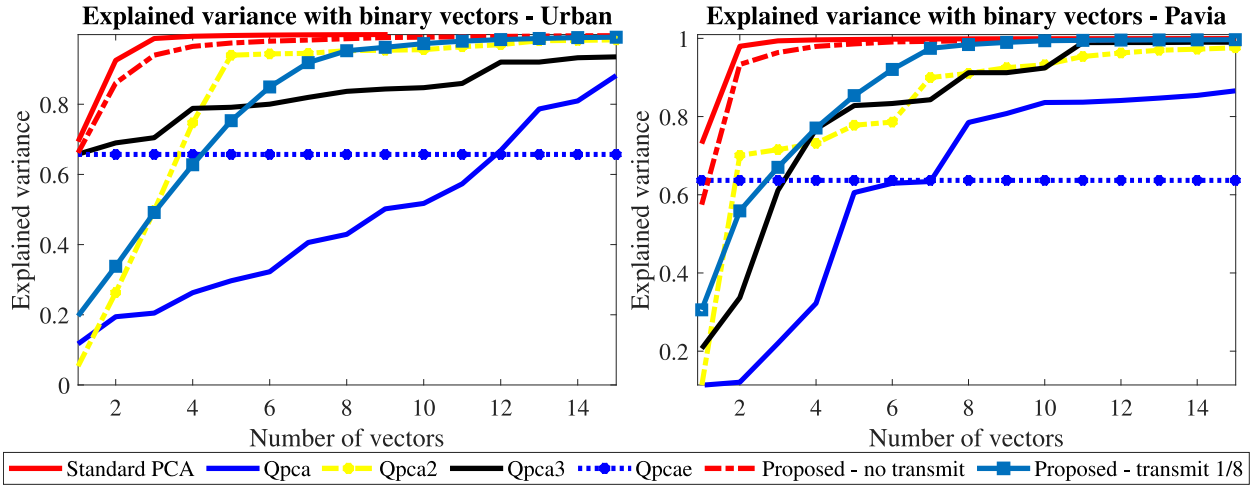


*Figure 16.* Explained variance with different realizations of Qpca. Qpca represents the explained variance when the pure eigenvectors are used; Qpca2 is obtained by scaling the eigenvectors with a diagonal matrix whose values are shown in Fig. 17(b); Qpca3 is obtained by scaling the eigenvectors with a diagonal matrix whose values are shown in Fig. 17(c); Qpcae is obtained by scaling the eigenvectors with the eigenvalues, which are shown in Fig. 17(d)

.

at the cost of rank deficiency as shown in Fig. 17.

In Figs. 16 and 17, it can be seen that by taking into account the importance of each eigenvector (determined by its associated eigenvalue) the Qpca algorithm converges to a solution where the variance is better explained. However, its convergence is not guaranteed since the rank of the resulting matrix degenerates as the matrix **B** resembles the eigenvalues. Note that, when the actual eigenvalues are used, the resulting binary matrix is rank 1, which negatively affects the conditioning of the problem. In contrast, in Figs. 16 and 12 it can be seen that the proposed method better explains the variance while the rank of the designed matrix does not degenerates.

**2.5.2. Binary Matrix vs. Discrete Matrix.** On the other hand, note that during the designing procedure, the sensing matrix is not binary at all, but discrete. However, once the algorithm finishes, the matrix is scaled to become binary. Hence, it should be highlighted that this

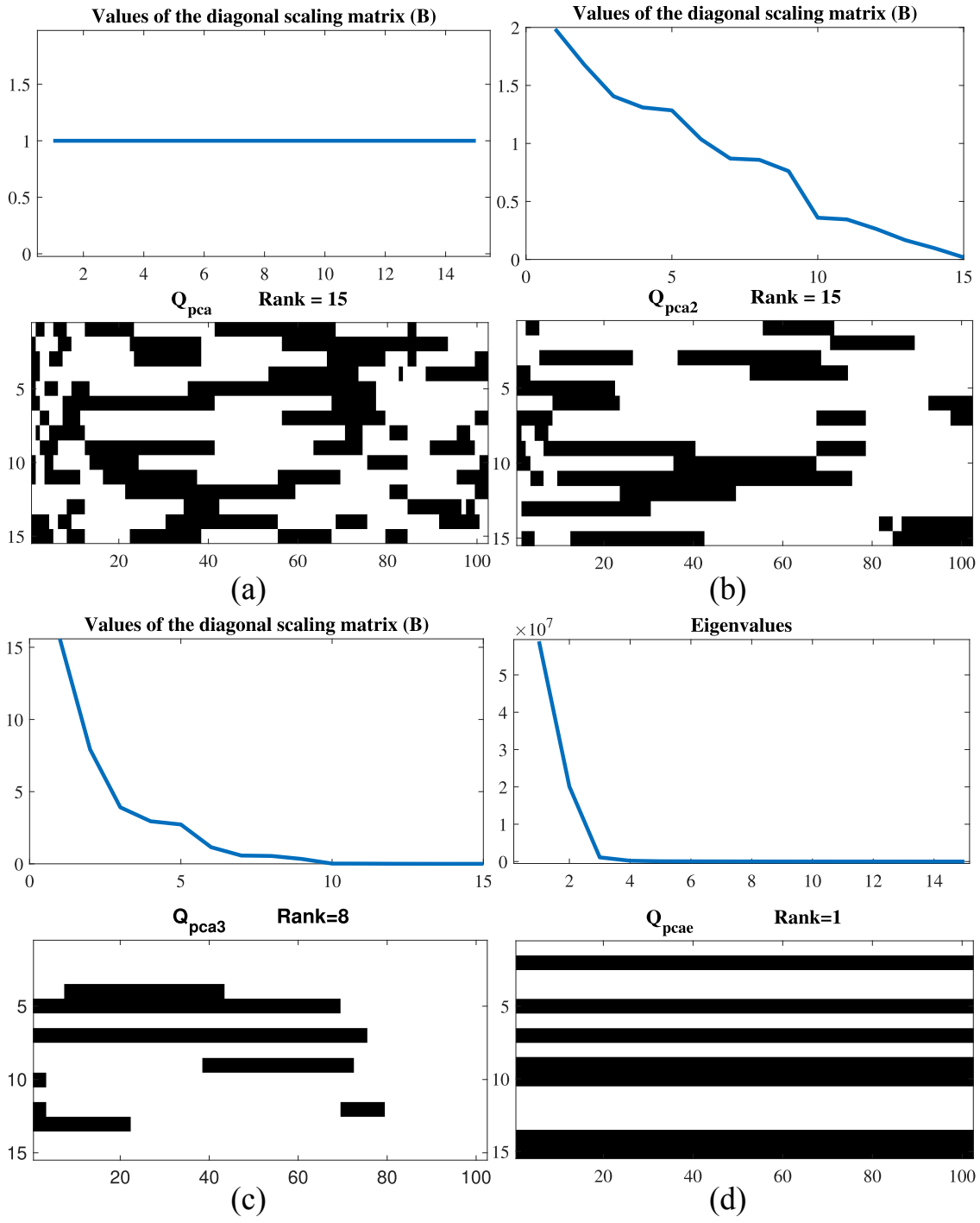


*Figure 17.* Values used in the diagonal matrix **B** to scale the eigenvectors of the Qpca algorithm, along with the resulting binary matrix. (a) Not scaling. (b) Smooth scaling. (c) Steep scaling. (d) Eigenvalue scaling.

scaling does not affect the resulting matrix, since an eigenvector defines a direction of projection, and so, any scaled version of an eigenvector is also an eigenvector. To see that, let's use the eigendecomposition of the covariance matrix

$$\mathbf{\Sigma} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^{-1}. \tag{46}$$

Hence, if $\mathbf{W}$ is scaled by a constant $b$, it holds that,

$$\mathbf{\Sigma} = b\mathbf{W}\mathbf{\Lambda}b^{-1}\mathbf{W}^{-1}, \tag{47}$$

is still an eigenvector. We expect the latter holds also for binary eigenvectors. To further support this affirmation, Fig. 46 shows that the explained variance is exactly the same when the binary or their discrete (normalized-column) version are used.

Furthermore, it can be seen in Fig. 46 of the supplementary material, that the explained variance obtained with the proposed binary eigenvectors is quite close to the standard PCA. This behavior is mainly due to hyperspectral images exhibit a low-rank behavior (Fowler, 2009). For a signal with not such behavior, the explained variance using binary eigenvectors is not that close to the standard, as shown and discussed in Fig. 47 of the supplementary material, where a random signal is compared against the Urban dataset.

### 2.6. Conclusions

This chapter introduced the design of binary sensing matrices, commonly used in real CSI architectures, via binary PCA. The designed matrices exploit the structure-preserving properties

of PCA, where most of the data variance is explained by a set of binary vectors, estimated directly from some compressed random measurements. The performance of the designed matrices was evaluated over two real hyperspectral datasets, Urban and Pavia center, achieving an overall 3 dB improvement in the reconstruction quality compared with conventional random sensing matrices and up to 2 dB compared with state-of-art sensing matrices based on PCA. Additionally, an analysis of the RIP for the proposed method was introduced, which provided a bound for the RIP when the standard PCA technique is performed and when the binary PCA is used. The proposed algorithm to design the matrices is greedy-search-based with low computational complexity, and the results show that it is able to retain variance of the data in a better fashion than state-of-art methods.

# 3. Reconstruction of the Covariance Matrix via A Projected Gradient Descend

*This chapter addresses the second and third objective of the thesis:*

- *To design an algorithm based on the gradient descent method to recover the first and second sample statistical moments from low-dimensional random projections.*
- *To test the performance of the proposed algorithm to recover the sample statistics in hyperspectral imaging reconstruction*

## 3.1. Introduction

The previous chapter addressed the problem of using the CM to guide the sensing process in CSI. When the covariance matrix is known beforehand, the algorithm performance is optimal. However, in CSI, only a set of compressed measurements is known, and reconstructing the signal to estimate the covariance matrix is computationally expensive. Hence, in chapter 2, we used the CPPCA algorithm (Fowler, 2009) to recover an approximation of the CM. CPPCA is a fast algorithm but suffers from multiple problems when applied to CSI. CPPCA requires that the sensing matrices be orthonormal, which can not be easily achieved in an optical implementation. Additionally, the CM eigenvalues must follow an eccentric distribution maximizing the separation between each other. Finally, the mathematical formulation of CPPCA does not consider the signal noise reducing the performance in real applications.

There exist also other algorithms for CM estimation reported for hyperspectral imaging. These methods have been used in different applications such as image reconstruction (Fowler and

Du, 2011; Li et al., 2013b), anomaly detection (Fowler and Du, 2012), and image classification (Li et al., 2013a). For example, the SpeCA approach introduces a spectral image recovery algorithm tailored to a particular sensor (Martin and Bioucas-Dias, 2016). More precisely, this method recovers the principal components of images by using a linear mixture model. Notice that the reported sensor requires sensing the entire image before obtaining the random compressive projections. Bioucas et al. proposed COVALSA (Bioucas-Dias et al., 2014b), an algorithm based on the ADMM approach that estimates the covariance matrix from compressive measurements assuming different structures such as Toeplitz, sparseness, and low rank. However, this method approximates the inverse function to use the ADMM approach adding extra hyperparameters. Our approach estimates the covariance matrix from compressive samples without using assumptions about the PCA coefficients compared to previous methods. Furthermore, the proposed approach can estimate the covariance matrix for a broader range of sampling operators, including random projections and CSI samples. In contrast to CPPCA and SpeCA, our method is evaluated using real compressive measurements captured by a practical optical setup. In addition, we analytically obtain the optimal number of partitions that recovers a reliable estimation.

**Chapter contribution.** This chapter develops an algorithm based on the projected gradient method to estimate the covariance matrix from compressive measurements. To this end, compressive measurements are divided into data subsets and projected onto multiple subspaces to improve the condition of the problem. Expressly, the estimation problem aims at recovering a low-rank or Toeplitz representation of a positive semidefinite matrix that minimizes the Frobenius norm of the projection errors. The proposed algorithm is evaluated for estimating the covariance ma-

trix embedded in hyperspectral image signatures using different compressive acquisition schemes, including random projections and binary encoding. It should note that, although the proposed method has been mainly tested on compressive samples derived from hyperspectral images, it can be extended to other image processing and communications applications. The contributions of this paper are summarized as follows: i) This paper proposes an optimization problem and a projected gradient-based covariance estimation method from compressive measurements using an Armijo search strategy to speed up convergence. The proposed method splits compressive samples into partitions projected on different subspaces to improve the estimation accuracy. The lower bound of the optimal number of partitions to obtain a reliable covariance matrix estimation is also derived (Lemma 2). ii) Moreover, this work derives theoretical guarantees for the global convergence of the proposed algorithm and determines the error term induced by the data splitting approach (Lemma 1). Likewise, a filtering strategy is proposed to mitigate the error induced by this error. iii) Finally, an implementable sensing protocol based on the DD-CASSI optical architecture is proposed and tested in the lab.

**3.1.1. Chapter organization.** The chapter is organized as follows: Section 3.2 introduces the covariance matrix estimation problem from random projections. Section 3.3 presents the optimization problem to be solved and the proposed algorithm for estimating covariance matrices from compressive projections in multiple subspaces. Sections 6 and 3.4 includes the global convergence guarantee of the proposed algorithm along with the error analysis. In Section 3.5, the performance of the proposed algorithm is evaluated using extensive numerical simulations using hyperspectral images. Additionally, an optical implementation is proposed to validate the thoreti-

cal findings. Some concluding remarks are summarized in Section 3.8.

### 3.2. Compressive covariance sampling formulation

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ be a matrix whose columns $\mathbf{x}_j \in \mathbb{R}^l$ for $j = 1, 2, \ldots, n$, are independent realizations of a zero-mean Gaussian random vector with covariance matrix $\mathbf{\Sigma}$ i.e., the distribution of $\mathbf{x}$ conditioned to $\mathbf{\Sigma}$ is

$$f(\mathbf{x}|\mathbf{\Sigma}) = \pi^{-l/2}|\mathbf{\Sigma}|^{-l/2}\text{etr}\left(\frac{1}{2}\mathbf{\Sigma}^{-1}\mathbf{x}\mathbf{x}^T\right), \tag{48}$$

where etr$(.)$ denotes the exponential of the trace. Under this context, the maximum likelihood estimator (MLE) for the covariance matrix reduces to the sample covariance matrix given by

$$\mathbf{S} = \frac{1}{n}\mathbf{X}\mathbf{X}^T = \frac{1}{n}\sum_{j=1}^{n}\mathbf{x}_j\mathbf{x}_j^T, \tag{49}$$

where $\mathbf{\Sigma} = \mathbb{E}[\mathbf{S}]$, $\mathbf{\Sigma} \in S_{++}^{l\times l}$, with $\mathbb{E}[\cdot]$ denoting the statistical expectation and $S_{++}^{l\times l}$ represents the set of positive definite matrices of size $l \times l$. However, in many practical applications, lower-dimensional signal projections are available instead of the target high-dimensional signal. In this regard, the sampling process that obtains lower-dimensional signal projections can be modeled as

$$\mathbf{Y} = \mathbf{P}^T\mathbf{X} + \mathbf{N} = [\mathbf{P}^T\mathbf{x}_1, \cdots, \mathbf{P}^T\mathbf{x}_2] + \mathbf{N}, \tag{50}$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n] \in \mathbb{R}^{m\times n}$ is the matrix containing the compressive projections $\mathbf{y}_j \in \mathbb{R}^m$ for $j = 1, 2, \ldots, n$, $\mathbf{P} \in \mathbb{R}^{l\times m}$ with $m < l$ represents the projection matrix; and $\mathbf{N} \in \mathbb{R}^{m\times n}$ is the additive noise matrix whose entries are characterized as independent and identically distributed (iid)

random samples following a zero-mean Gaussian model with variance $\sigma_N^2$, i.e. $N_{i,j} \sim \mathcal{N}(0, \sigma_N^2)$. Notice that the sample covariance matrix obtained from the observation matrix is obtained as

$$\tilde{\mathbf{S}} = \frac{1}{n}\mathbf{Y}\mathbf{Y}^T = \frac{1}{n}(\mathbf{P}^T\mathbf{X} + \mathbf{N})(\mathbf{P}^T\mathbf{X} + \mathbf{N})^T, \tag{51}$$

with $\tilde{\mathbf{S}} \in \mathbb{R}^{m \times m}$. Since $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ and $\mathbf{P}$ is a fixed matrix, the projected vectors $\{\mathbf{y}_j\}_{j=1}^n$ can be modeled as zero-mean Gaussian vectors with covariance given by $\mathbf{P}^T\mathbf{\Sigma}\mathbf{P} + \sigma_N^2\mathbf{I}$, i.e. $\mathbf{y} \sim \mathcal{N}(0, \mathbf{P}^T\mathbf{\Sigma}\mathbf{P} + \sigma_N^2\mathbf{I})$. Furthermore, it can be observed that $n\tilde{\mathbf{S}}$ follows a Wishart distribution, that is, $n\tilde{\mathbf{S}} \sim \mathcal{W}(\mathbf{P}^T\mathbf{\Sigma}\mathbf{P} + \sigma_N^2\mathbf{I}, n)$(Besson et al., 2008a).

The above assumptions lead to the minimization of the Frobenius norm of the residuals between the covariance matrix of the projected vectors and the projected version of the covariance matrix estimate as the optimal performance criterion. However, this approach leads to ill-posed optimizations with significant performance losses at high compression rates. To overcome this limitation, a regularization term is aggregated to the cost function based on a particular covariance matrix structure, e.g., low-rank or Toeplitz. The optimization problem to recover the sample covariance matrix $\mathbf{\Sigma}$ from $\tilde{\mathbf{S}}$ is formulated as (Bioucas-Dias et al., 2014b)

$$\mathbf{\Sigma}^* = \underset{\mathbf{\Sigma} \in \mathbf{D}}{\operatorname{argmin}} \quad \|\tilde{\mathbf{S}} - \mathbf{P}^T\mathbf{\Sigma}\mathbf{P}\|_F^2 + \tau\psi(\mathbf{\Sigma}), \tag{52}$$

where $\psi(\cdot)$ is a convex function that regularizes the problem, $\tau$ is the regularization parameter, $\|\cdot\|_F$ denotes the Frobenius norm, and $\mathbf{D}$ is a proper convex and closed set, e.g., the set of positive

semi-definitive or Toeplitz matrices.

Moreover, note that the zero-mean assumption in (49) does not hold in different image processing applications. Hence, the random projections can be alternatively written as:

$$\mathbf{Y} = \mathbf{P}^T(\mathbf{X} + \tilde{\mathbf{X}}) + \mathbf{N}, \tag{53}$$

where $\tilde{\mathbf{X}} = \tilde{\mathbf{x}}\mathbf{1}^T$ is a matrix whose columns are the mean vector, i.e., $\tilde{\mathbf{x}} = \mathbb{E}[\mathbf{x}]$, and $\mathbf{1} \in \mathbb{R}^n$ is an $n$-dimensional vector with one-valued entries. Notice that an estimate of the mean vector can be obtained from the compressive projections (Qi and Hughes, 2012; Anaraki and Hughes, 2014) as follows

$$\tilde{\mathbf{x}} = \alpha \sum_{j=1}^{n} \mathbf{P}(\mathbf{P}^T\mathbf{P})^{-1}\mathbf{y}^j, \tag{54}$$

where $\alpha = m/n$ and $\mathbf{y}^j$ is the *j-th* vector in $\mathbf{P}^T\mathbf{X}$. It has been proved in (Qi and Hughes, 2012) that (54) converges to the mean vector when $n \to \infty$. Once the mean is estimated, the measurements can be corrected by subtracting the projection of the estimated mean vector to the biased samples, i.e., $\tilde{\mathbf{Y}} = \mathbf{Y} - \mathbf{P}^T(\widehat{\mathbf{x}}\mathbf{1}^T)$. Without loss of generality, we assume that signals are zero mean.

### 3.3. Recovery of the covariance matrix from compressed measurements

Solving (52) typically yields poor results at high compression ratios of the projection vectors $m/l$. This behavior is attributed to all vectors being projected onto a single subspace and possibly projected onto the null space. Hence, we split the data into disjoint subsets projected onto different subspaces to improve the performance of the estimator. The partitioning into multiple subsets has been previously used for the CPPCA sensing approach(Fowler, 2009). However, this

approach requires that the sensing matrices be orthonormal.

**3.3.1. Projection set up and optimization problem.** Let's split the dataset $\mathbf{X}$ into $p$ disjoint subsets, $\mathbf{X}_i$, with columns defined as $\mathbf{X}_i = [\mathbf{x}_{\Omega_{i1}}, \cdots, \mathbf{x}_{\Omega_{ib}}]$ with $i = 1, 2, \cdots, p$ and $\Omega_{ij} = \Omega_{i'j'}$ only if $i = i'$ and $j = j'$. Since each column $\mathbf{x}_{\Omega_{i1}} \sim \mathcal{N}(0, \mathbf{\Sigma})$, it holds that the sample covariance matrix of every subset $b\mathbf{S}_i = \mathbf{X}_i\mathbf{X}_i^T \sim \mathcal{W}(\mathbf{\Sigma}, b)$, where $b = n/p$ is the number of columns in each subset. Then, each subset $\mathbf{X}_i \in \mathbb{R}^{l\times b}$ is projected in a lower-dimensional subspace with an independent matrix $\mathbf{P}_i \in \mathbb{R}^{l\times m}$, this is,

$$\mathbf{Y}_i = \mathbf{P}_i^T\mathbf{X}_i + \mathbf{N}_i. \tag{55}$$

Using this splitting procedure, each $\mathbf{S}_i$ matrix can be estimated solving the optimization (52), in other words,

$$\mathbf{S}_i^* = \underset{\mathbf{S}_i\in\mathbf{D}}{\operatorname{argmin}} \quad ||\tilde{\mathbf{S}}_i - \mathbf{P}_i^T\mathbf{S}_i\mathbf{P}_i||_F^2 + \tau\psi(\mathbf{S}_i) \tag{56}$$

where $\tilde{\mathbf{S}}_i = \frac{1}{b}\mathbf{Y}_i\mathbf{Y}_i^T$ for $i = 1, \cdots, p$. Notice that the formulation in (56) involves $p$ different optimization problems, one for each matrix $\mathbf{S}_i$, which increases the number of unknowns and thus the ill-posedness of the problem. However, note that for a sub-Gaussian process, it holds that

$$||\mathbf{S}_i - \mathbf{\Sigma}|| \leq \varepsilon, \tag{57}$$

with probability at least $1 - 2\exp(-t^2 l)$, for $b \geq J(t/\varepsilon)^2 l$, $\varepsilon \in (0,1), t \geq 1$, and $J$ depends on the sub-Gaussian norm of $\mathbf{X}$(Vershynin, 2010), i.e. the statistics of a subset can approximately describe

the statistics of the whole random process.

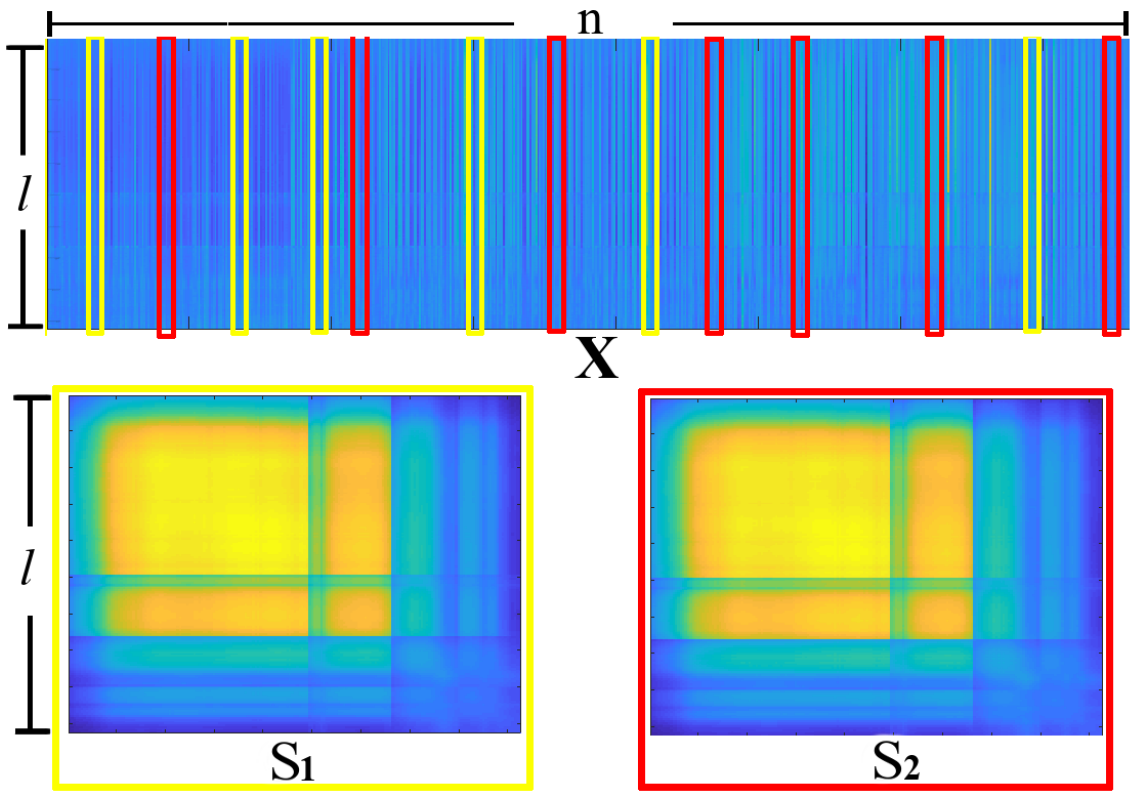


*Figure 18.* The partition approach. For a matrix $\mathbf{X}$, different subsets of columns are selected, and their covariance matrices are computed. The matrix $\mathbf{S}_1$ represents the covariance matrix of the yellow columns, and $\mathbf{S}_2$ represents the covariance matrix of the red columns.

Fig. 18 illustrates the similarity when the covariance matrices of two subsets of the signal $\mathbf{X}$ are presented. To that end, two subsets of columns of the matrix $\mathbf{X}$ highlighted in yellow and red are used to compute the covariance matrices $\mathbf{S}_1$ and $\mathbf{S}_2$. This example $\mathbf{X}$ is a matrix representation of a hyperspectral image with a spatial resolution of $512 \times 512$ (i.e. $n = 262144$) and $l = 102$ spectral bands, where each column of $\mathbf{X}$ represents the spectrum at a given spatial location. The computation of the matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ uses $b = 2048$ spectral signatures. As it can be seen, these two matrices are similar as $||\mathbf{S}_1 - \mathbf{S}_2||_F = 0.0321$. Instead of recovering all covariance matrices $\mathbf{S}_i$,

we assume that $\mathbf{S}_1 = \mathbf{S}_2 = \cdots = \mathbf{S}_p = \mathbf{\Sigma}$. By doing this, we merge the separate problems in (56) into a single optimization problem by replacing $\mathbf{S}_i$ with $\mathbf{\Sigma}$, which results in

$$\mathbf{\Sigma}^* = \underset{\mathbf{\Sigma}\in\mathbf{D}}{\operatorname{argmin}} \quad \sum_{i=1}^{p} ||\tilde{\mathbf{S}}_i - \mathbf{P}_i^T \mathbf{\Sigma} \mathbf{P}_i||_F^2 + \tau\psi(\mathbf{\Sigma}). \tag{58}$$

Theoretical results and simulations show that this splitting procedure improves the accuracy and variance of the estimator (see Lemma 2). Additionally, an advantage of (58) in contrast to (52) is that even if an eigenvector falls in the null space of a given matrix $\mathbf{P}_k$, the probability that the eigenvector falls in the null space of every matrix $\mathbf{P}_i$ is small as $m \times p \geq l$, and thus the probability of correct reconstruction increases. To see that, consider a matrix $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2, \cdots, \mathbf{P}_p]^T \in \mathbb{R}^{mp\times l}$ whose entries are independent identically distributed subgaussian random variables with zero mean and unit variance. In general, the row null space of the matrix $\mathbf{P}$ is empty if the minimum singular value is greater than 0, i.e., $s_{min}(\mathbf{P}) > 0$. The probability that the minimum singular value is less than a small number is given by(Vershynin, 2010)

$$\mathbb{P}[\text{null}(\mathbf{P}) = \emptyset] \geq 1 - \mathbb{P}\left(s_{min}(\mathbf{P}) \leq \varepsilon(\sqrt{mp} - \sqrt{l-1})\right) = 1 - (C\varepsilon)^{mp-l+1} + c^{mp} \tag{59}$$

with $\varepsilon \geq 0$, $C > 0$, and $c \in (0,1)$. Note that the probability that the null space to be empty increases exponentially with respect to the subspace dimension $m$ and the number of partitions $p$. Hence, in the case of having a single partition $p$, the only way to increase $\mathbb{P}[\text{null} = \emptyset]$ is to increase the subspace dimension. In the hyperspectral imaging context, increasing $m$ implies acquiring more

snapshots reducing the compression. On the other hand, this can also be achieved by increasing the number of partitions $p$ while the compression remains constant.

**3.3.2. Proposed projected gradient algorithm for covariance matrix recovery.** Problem (58) is solved following the projected gradient method. This method requires a differentiable function $f(\mathbf{\Sigma})$ and a proper closed and convex set $\mathbf{D}$ to formulate the optimization problem as

$$\begin{aligned} \mathbf{\Sigma}^* = \underset{\mathbf{\Sigma}\in\mathbb{R}^{l\times l}}{\operatorname{argmin}} \quad & f(\mathbf{\Sigma}) \\ \text{subject to} \quad & \mathbf{\Sigma} \in D. \end{aligned} \tag{60}$$

Note that (58) has the form of (60), so it can be solved using the projected gradient algorithm as illustrated in Algorithm 3.1. This algorithm is summarized in three main steps. First, a starting point $\mathbf{\Sigma}^0 \in D$ and the regularization parameter $\tau$ are selected. Parameter $\tau$ induces the low-rank structure in the solution. Step two (line three), the learning step is selected using the Armijo search(Iusem, 2003). For that, $\lambda_k = \lambda_{k-1}/\eta^r$, where $\eta > 1$, $\lambda_{-1} > 0$ and $r$ is the smallest positive integer (including 0) that satisfies

$$\min_{r} f(\mathbf{\Sigma}^{k_r}, \tau) \leq f(\mathbf{\Sigma}^k, \tau) + \operatorname{Tr}(\nabla f(\mathbf{\Sigma}^k, \tau)^T(\mathbf{\Sigma}^{k_r})) + ||\mathbf{\Sigma}^{k_r} - \mathbf{\Sigma}^k||_F^2, \tag{61}$$

where $\mathbf{\Sigma}^k$ is the *k-th* iteration, $\mathbf{\Sigma}^{k_r} = P_{\mathbf{D}}(\mathbf{\Sigma}^k - (\lambda_{k-1}/\eta^r)\nabla f(\mathbf{\Sigma}^k, \tau))$ is an intermediate step between iterations $k$ and $k+1$ and $P_{\mathbf{D}}$ is the projection onto the set $\mathbf{D}$. In step 3 (line 4), the variable $\mathbf{\Sigma}_k$ is

updated by using the gradient of the cost function

$$f(\mathbf{\Sigma}, \tau) \equiv \sum_{i=i}^{p} ||\tilde{\mathbf{S}}_i - \mathbf{P}_i^T \mathbf{\Sigma} \mathbf{P}_i||_F^2 + \tau \psi(\mathbf{\Sigma}), \tag{62}$$

which for a fixed $\tau$ is given by

$$\nabla f(\mathbf{\Sigma}, \tau) = \sum_{i=1}^{p} \mathbf{P}_i (\tilde{\mathbf{S}}_i - \mathbf{P}_i^T \mathbf{\Sigma} \mathbf{P}_i) \mathbf{P}_i^T + \tau \nabla \psi(\mathbf{\Sigma}). \tag{63}$$

When $\psi(\mathbf{\Sigma}) = \mathrm{Tr}(\mathbf{\Sigma})$, which is used for low-rank structure, the gradient is given by $\nabla \mathrm{Tr}(\mathbf{\Sigma}) = \mathbf{I} \in \mathbb{R}^{l \times l}$. Once the variable $\mathbf{\Sigma}_k$ is updated using the gradient, it is projected onto the set $D$, whose computation depends on the set $D$ itself. This work studies two sets:

1. **Positive semi-definitive:** The orthogonal projection onto the set of positive semi-definitive matrices is given by(Grigoriadis et al., 1994)

$$P_D(\mathbf{\Sigma}) = \mathbf{W} \mathbf{\Lambda}_+ \mathbf{W}^T, \tag{64}$$

   where $\mathbf{W}$ is the matrix containing the eigenvectors and $\mathbf{\Lambda}_+$ is the matrix containing only the positive eigenvalues of $\mathbf{\Sigma}$.

2. **Toeplitz:** The orthogonal projection onto the set of Toeplitz matrices is given by(Grigoriadis

et al., 1994)

$$P_D(\mathbf{\Sigma}) = \begin{bmatrix} t_0 & t_1 & t_2 & \dots & t_{l-1} \\ t_1 & t_0 & t_{-1} & \dots & t_{l-2} \\ t_2 & t_1 & t_0 & \dots & t_{l-3} \\ \dots & \dots & \dots & \dots & \dots \\ t_{l-1} & t_{l-2} & t_{l-3} & \dots & t_0 \end{bmatrix}, \tag{65}$$

where $t_k = 1/(n-k)\sum_{i=1}^{n-k}\Sigma_{i,(i+k)}$, with $\mathbf{\Sigma} = \{\Sigma_{i,j}\}$.

Thus, the proposed gradient algorithm can be summarized by Alg. 3.1.

**Algorithm 3.1** Projected gradient algorithm

1: $\mathbf{\Sigma}^0 \in \mathbf{D}, \tau, \lambda_0$

2: **while** stopping criteria is not satisfied **do**

3: pick $\lambda_k > 0$ ▷ Armijo search

4: $\mathbf{\Sigma}^{k+1} \leftarrow P_{\mathbf{D}}(\mathbf{\Sigma}^k - \lambda_k \nabla f(\mathbf{\Sigma}^k, \tau))$ ▷ Using 1) or 2)

5: **end while**

Note that, Algorithm 3.1 works for both low-rank and Toeplitz cases. Nevertheless, for the Toeplitz case, $\tau$ is set to zero since the low-rank constraint is unnecessary. The algorithm convergence analysis is presented in the Supplementary material, Section 6

### 3.4. Error term of the proposed estimator

The assumption in (57) introduces an error term in the gradient. To show that, let us characterize the difference of the ground-truth sample covariance matrix $\mathbf{S}$ and the covariance matrices

$\mathbf{S}_i$ as

$$\mathbf{S}_i = \mathbf{S} + \mathbf{R}_i, \tag{66}$$

where $\mathbf{R}_i \in \mathbb{R}^{l\times l}$ is a matrix that accounts for the error between the covariance matrices. In the ideal case, where $\mathbf{S} = \mathbf{S}_i \forall i$, the estimator is optimal and (63) holds. However, in the more realistic scenario where $\mathbf{S} \neq \mathbf{S}_i \forall i$, assuming (66), the error is described in lemma 1

**Lemma 1.** The gradient step for the proposed Algorithm 3.1 has an error term given by $\text{Error}[\nabla \tilde{f}(\mathbf{\Sigma})] = -\sum_{i=1}^{p} \mathbf{P}_i \mathbf{P}_i^T \mathbf{R}_i \mathbf{P}_i \mathbf{P}_i^T$.

where $\text{Error}[\cdot] = \nabla f - \nabla \tilde{f}$, and $\nabla f, \nabla \tilde{f}$ are the optimal and actual gradients respectively.

*Proof:* See Appendix 3.6.

An important property of the error term is that it is proportional to the number of subsets, and thus the error associated with $\text{Error}[\nabla \tilde{f}(\mathbf{\Sigma})]$ increases with the number of partitions $p$. However, more partitions improve the condition of the information matrix of the problem. Consequently, choosing the number of subsets is a trade off between improving the condition of the problem and increasing the error. The following theorem bounds the latter.

**Theorem 2.** The variance for any Covariance matrix $\mathbf{\Sigma}$ estimator for (55) with deterministic projection matrices $\mathbf{P}_i$, assuming that is non-singular, satisfies

$$\text{var}(\tilde{\mathbf{\Sigma}}) \geq \frac{p}{n} \text{Tr}\left[ \left( \sum_{i=1}^{p} \mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T \right)^{-1} \right], \tag{67}$$

with $\mathbf{A}_i = (\mathbf{\Sigma}_N + \mathbf{P}_i^T \mathbf{\Sigma} \mathbf{P}_i)^{-1}$, and $\left(\sum_{i=1}^{p} \mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T\right)$ is the information matrix. *Proof:* See

appendix 6.

From Theorem 2 it is important notice that for small values of $p$ the fisher information matrix is singular. Hence, a large enough number of partitions ($p > l^2/m^2$) must be performed based on lemma 2

**Lemma 2.** Let $\mathbf{P}_i \in \mathbb{R}^{l\times m}$ and $\mathbf{A}_i \in \mathbb{R}^{m\times m}$, then the matrix $\sum_{i=1}^{p} \mathbf{P}_i\mathbf{A}_i^T\mathbf{P}_i^T \otimes \mathbf{P}_i\mathbf{A}_i\mathbf{P}_i^T$ is singular if $p < l^2/m^2$. *Proof:* See appendix 6

From Lemma 2, it can be seen that the information matrix is non-singular for some $d > 1$ such that $p \geq dl^2/m^2$. Nevertheless, choosing large $p$ increase the norm of the error term given in Lemma 1, as shown in (141). Hence, $p$ should be chosen big enough to avoid the singularity of (67) but small enough to decrease the error term in (75), which yields an optimal number of partitions of $p = \lceil (l^2/m^2) + 1 \rceil$. Additionally, this error term follows an important property given by lemma 3

**Lemma 3.** Let $\{\mathbf{R}_i\}$ be the set of error matrices for the subsets covariance matrices $\mathbf{S}_i$, hence since the sensing matrices $\mathbf{P}_i$ are deterministic and $\mathbb{E}[\mathbf{R}] = \mathbf{0}$ (Appendix 6), for any entry $\mathbf{B}_{\iota\rho}$ of the matrix $\mathbf{P}_i\mathbf{P}_i^T\mathbf{R}_i\mathbf{P}_i\mathbf{P}_i^T = \mathbf{B}$ it holds that

$$\mathbb{E}[\mathbf{B}_{\iota\rho}] = 0, \tag{68}$$

*Proof:* See Appendix 6.

This result motivates the use of a filtered gradient to remove the effect of the error term. Simulations show that this error is usually associated with high frequencies. Moreover, the proposed algorithm filters the gradient in each iteration to mitigate this error, especially when the

compression is high since more partitions are required (as can be seen in Lemma 2). The filtered gradient is given by

$$\nabla \hat{f}(\mathbf{\Sigma}) = \mathbf{K} * \nabla f(\mathbf{\Sigma}), \tag{69}$$

where $*$ represents the convolution operation, and $\mathbf{K} \in \mathbb{R}^{k \times k}$ is the filter kernel. This new gradient is used in step 4 of algorithm 3.1. This filtering step reduces the error term variance, as shown in Appendix 6 in the supplementary material. Additionally, the norm of the error term is bounded by:

$$\mathbb{P}\left\{ \left\| \sum_{i=1}^{p} \mathbf{H}_i \mathbf{R}_i \mathbf{H}_i \right\|_2 \geq t \right\} \leq 2 \times l \times e^{\frac{-t^2/2}{p^2\sigma_H^2 + \sigma_m^2 \varepsilon t/3}}, \tag{70}$$

for all $t \geq 0$, $\sigma_m = \max(\sigma_{max}(\mathbf{H}_i))$ and $\sigma_H = \sigma_m^4 \varepsilon^2$. *Proof:* See Appendix 6 in supplementary material.

### 3.5. Simulations and Results

The performance of the proposed algorithm is tested using synthetic and real data. The gradient is filtered using a Gaussian filter with $\sigma = 1$; however, it is only used along with the low-rank restriction (i.e., $\tau > 0$). Three different projection matrices $\mathbf{P}$ are used: *i)* Gaussian matrices whose entries follow a standard normal distribution $\mathbf{P}_{i,j} \sim \mathcal{N}(0,1)$; *ii)* Binary matrices with entries $\mathbf{P}_{i,j} \sim$ Bernoulli$(p = \frac{1}{3})$; and *iii)* matrices whose elements obey to a standard uniform distribution, $\{\mathbf{P}\}_{i,j} \sim U(0,1)$. In simulations, two noisy scenarios of 20 and 30 dB SNR were tested with SNR defined as SNR=$10\log ||\mathbf{P}^T\mathbf{X}||_F^2 / ||\mathbf{N}||_F^2$.

**3.5.1. Synthetic data performance evaluation.** Synthetic data from a low-rank and Toeplitz covariance matrices were generated. For the low-rank covariance matrix, the data

points were generated using Matlab with $\boldsymbol{\mu} = 0$, the rank of $\boldsymbol{\Sigma}$ set to 7, and the dimension of the signal $l = 100$. The data from the Toeplitz matrix was generated as an autoregressive model of order $q = 8$ and dimension of the signal $l = 100$. For the reconstruction algorithm $\tau = \rho * \text{trace}(\mathbf{S}_0)$, where $\mathbf{S}_0$ is the initialization of the covariance matrix, and $\rho$ was chosen using cross-validation. More details are available in the supplementary material.

Fig. 19 shows the average normalized mean squared error defined as NMSE=$||\boldsymbol{\Sigma} - \tilde{\boldsymbol{\Sigma}}||_F / ||\boldsymbol{\Sigma}||_F$ as a function of the number of partitions between the original and reconstructed covariance matrices. It can be seen that Gaussian matrices have the best performance. Based on those results, we set the number of partitions to 4 and 128 for Toeplitz and low-rank data, respectively in synthetic data experiments. The proposed algorithm results are compared against sparse rulers and a least squares autoregressive estimator for the Toeplitz matrix. The proposed algorithm is compared against the compressive-projection principal component analysis (CPPCA)(Fowler, 2009) and the spectral compressive acquisition (SpeCA) method for the low-rank matrix(Martin and Bioucas-Dias, 2016).

Figure 20 shows that both the proposed and SpeCA algorithms outperform the CPPCA algorithm because the generated random signal does not exhibit an eccentric behavior in the eigenvalues of the covariance matrix which is an essential assumption for the CPPCA algorithm. On the other hand, the proposed algorithm achieves comparable results to the SpeCA when Gaussian matrices are used but outperforms the SpeCA with binary matrices and in low SNR regimes.

Figure 21 compares the performance of different algorithms in the recovery of the Toeplitz covariance matrix. The proposed algorithm outperforms two state-of-the-art algorithms, AR co-

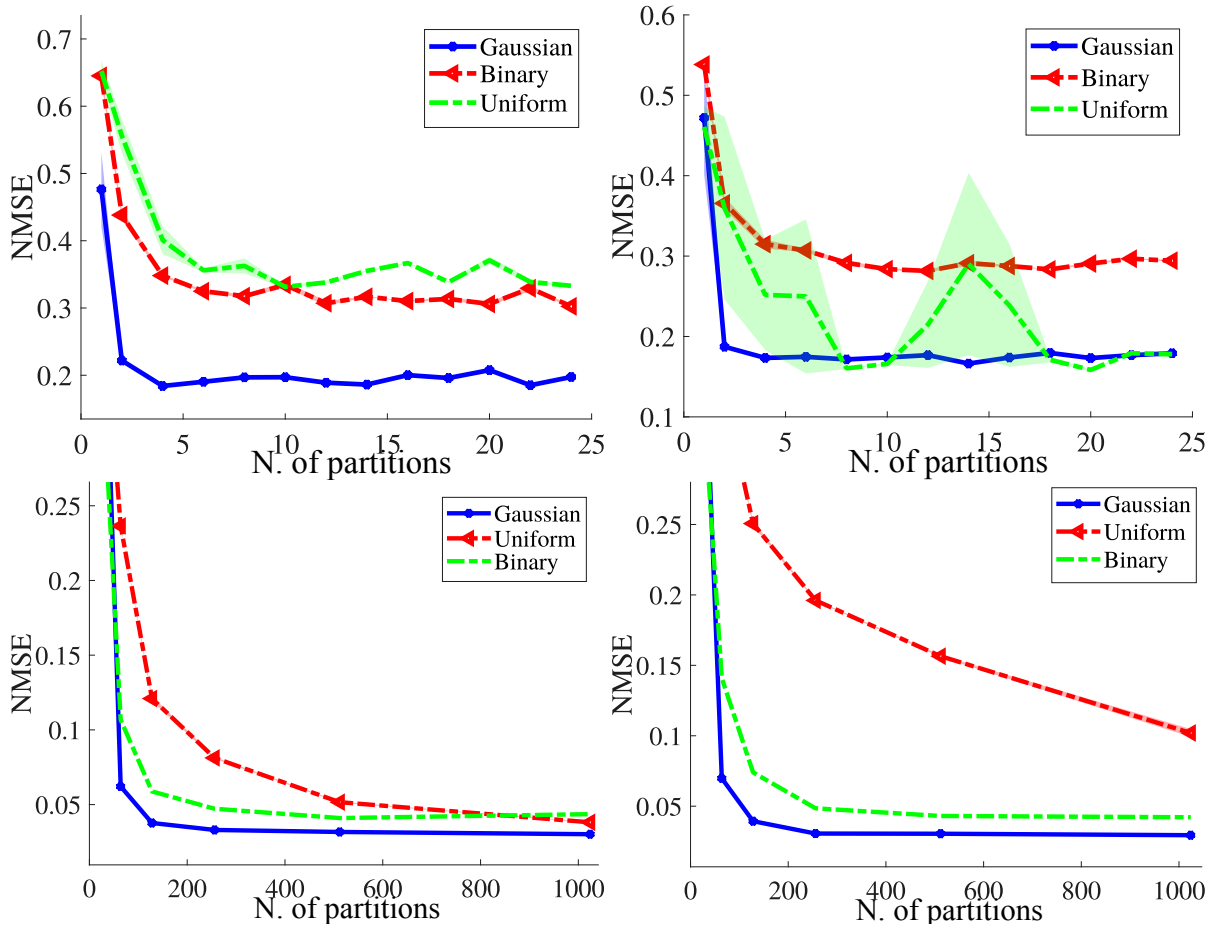


*Figure 19.* Average normalized mean square error of the reconstructed covariance matrix for the (top) Toeplitz and (bottom) low-rank matrices varying the number of partitions using 8% of compression ratio with two noise scenarios (left) SNR=30dB and (right) SNR=20dB. Each line represents a different sensing matrix (Gaussian, Uniform, Binary). The shaded areas represent the confidence interval (in some cases, it can not be seen in the plot).

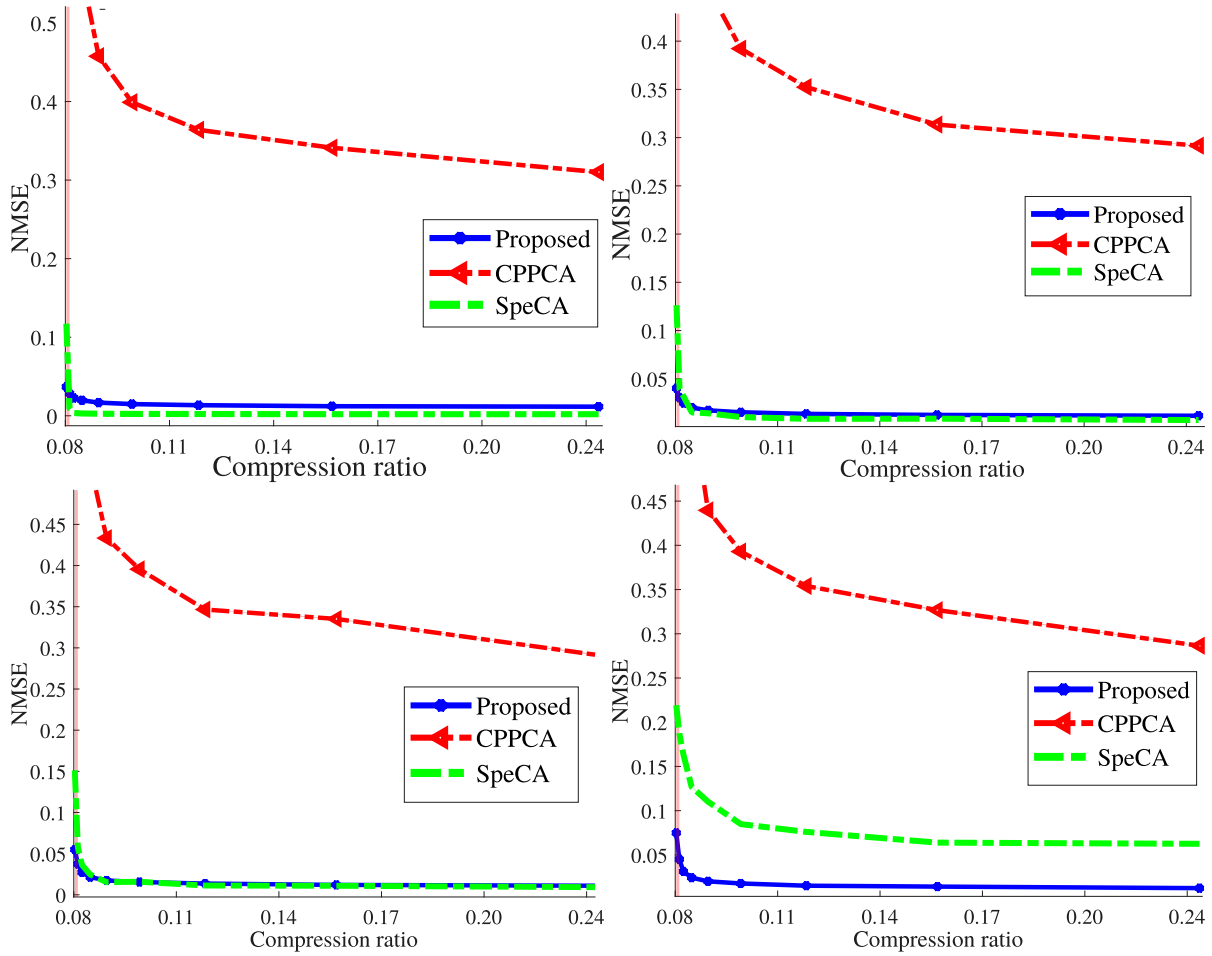


*Figure 20.* Average normalized mean square error of the reconstructed covariance matrix for the low-rank covariance matrix varying the number of acquisitions using 128 partitions with two noise scenarios (left) SNR=30dB and (right) SNR=20dB and two types of sensing matrices, Gaussian (top) and Binary(Bottom)

efficient(Testa and Magli, 2016), and Sparse rulers(Romero et al., 2016b), especially with high compression ratios. The proposed method is compared using two sensing matrices, Gaussian and Binary. Note that both AR coefficients and sparse rulers propose a specific sensing protocol, and hence the sensing matrix is fixed.

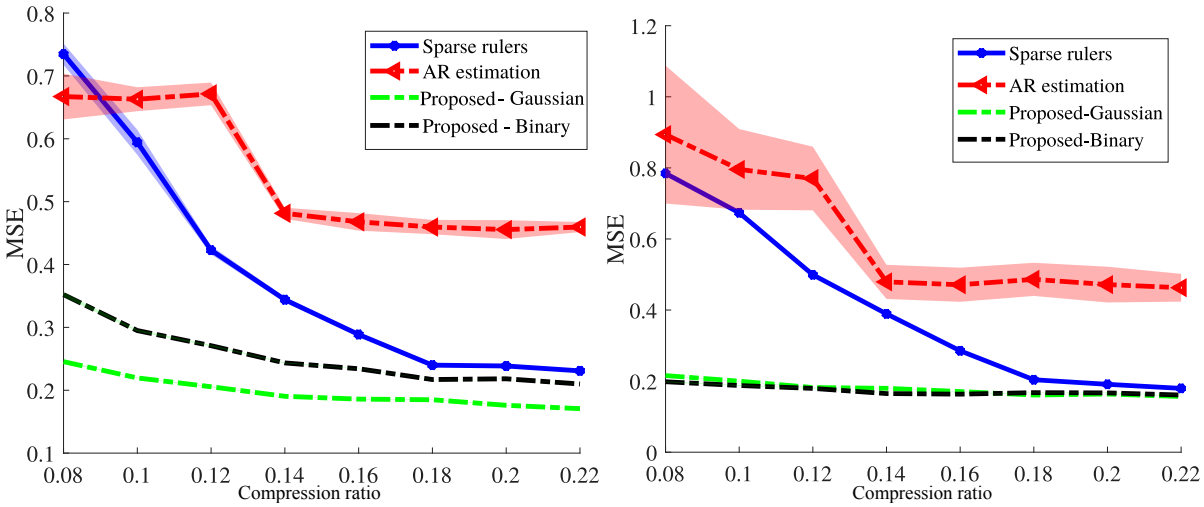


*Figure 21.* Average normalized mean square error of the reconstructed covariance matrix for the Toeplitz covariance matrix varying the number of acquisitions using four partitions with two noise scenarios (left) SNR=30dB and (right) SNR=20dB. The shaded areas represent the confidence interval.

**3.5.2. Computational simulations with Hyperspectral images.** Additionally, the proposed method is evaluated by estimating the covariance matrix of hyperspectral images using subsets of random compressive projections. Two hyperspectral images are considered: the Urban dataset (Zhu et al., 2014) with a spatial resolution of $256 \times 256$ pixels and $l = 128$ spectral bands; and a section of the Pavia Centre dataset (Mueller et al., 2002) with dimensions $512 \times 512 \times 102$. The RGB composite and the spectral signatures of three pixels (at the spatial locations P1, P2, and P3) for the Urban dataset are displayed in Fig. 22 (Top-Left) and (Top-Right), respectively. Moreover, Figs. 22(Bottom-Left)-(Bottom-Right) show the RGB composite and the spectral signatures for the Pavia Centre dataset. The results obtained with the proposed method are compared with those obtained using the CPPCA and the SpeCA algorithms. Three metrics are used to compare the results, the Mean Square Error (MSE) between the covariance matrices, the error angle be-

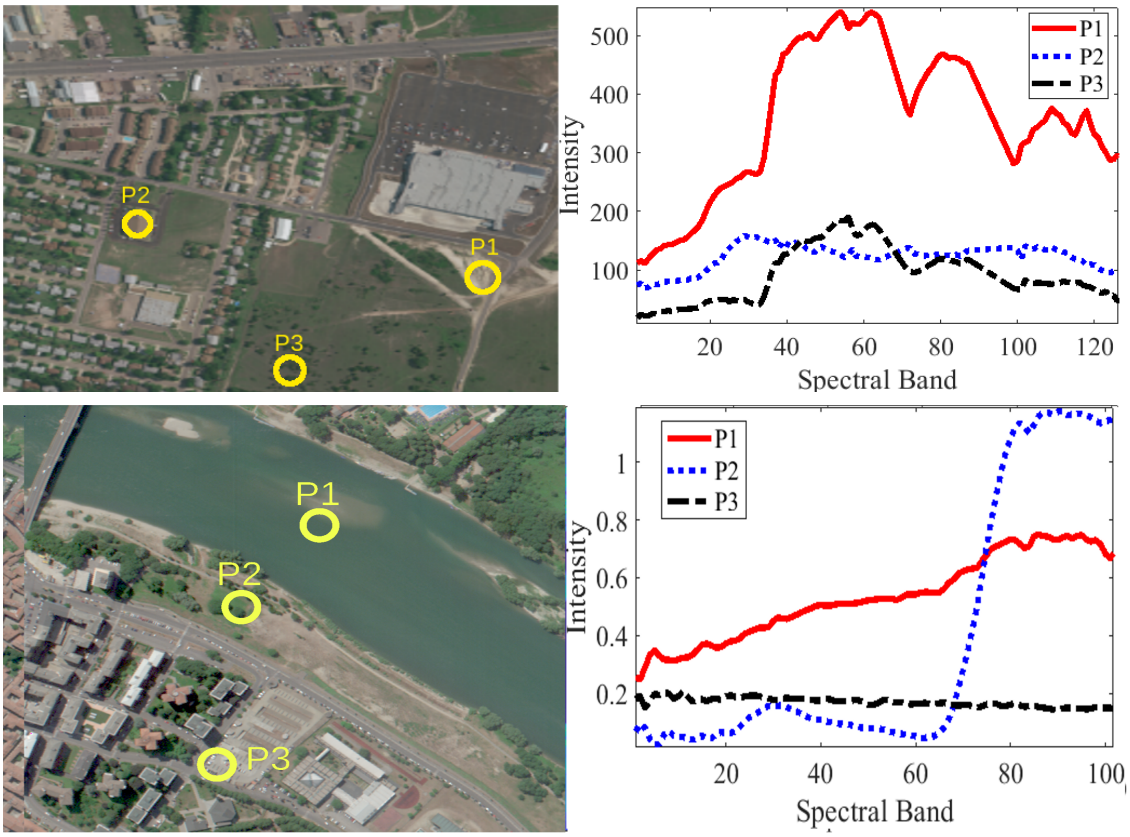


*Figure 22.* (Top) Urban dataset: (Left) RGB composite of the hyperspectral image, (Right) spectral signatures of three different pixels at locations P1, P2, and P3.(Bottom) Pavia Centre dataset: (Left) RGB composite of the spectral image, (Right) spectral signatures of three different pixels.

tween the eigenvectors, and the Peak Signal to Noise Ratio (PSNR). The sample covariance matrix $\mathbf{S} = \mathbf{X}\mathbf{X}^T/n$ is used as the truth covariance matrix for the simulations.

**3.5.3. Cramer-rao lower bound and optimal number of partitions.** As described in Section 3.3, the signal splits into $p$ subsets projected using different matrices and $p \geq l^2/m^2$ as described by Lemma 2. This section evaluates the estimator's variance using the theoretical expression given in Theorem 2 and the empirical variance in the simulations. Table 4 shows the value $l^2/m^2$ for both images as $m$ increases. Fig. 23 shows the theoretical variance given

Table 4
*Minimum optimal number of partitions.*

| image/m | 8 | 12 | 16 | 20 | 24 | 28 | 32 |
|---|---|---|---|---|---|---|---|
| **Urban** $p$ | 256 | 114 | 64 | 41 | 29 | 21 | 16 |
| **Pavia** $p$ | 163 | 72 | 41 | 26 | 18 | 13 | 10 |

by the Cramer-rao lower bound and the empirical variance defined as $1/r\text{tr}[\sum_{i=1}^{r}(\tilde{\boldsymbol{\sigma}}-\boldsymbol{\sigma})(\tilde{\boldsymbol{\sigma}}-\boldsymbol{\sigma})^T]$ where $r$ is the number of realizations. Fig. 23 presents three different compression ratio scenar-

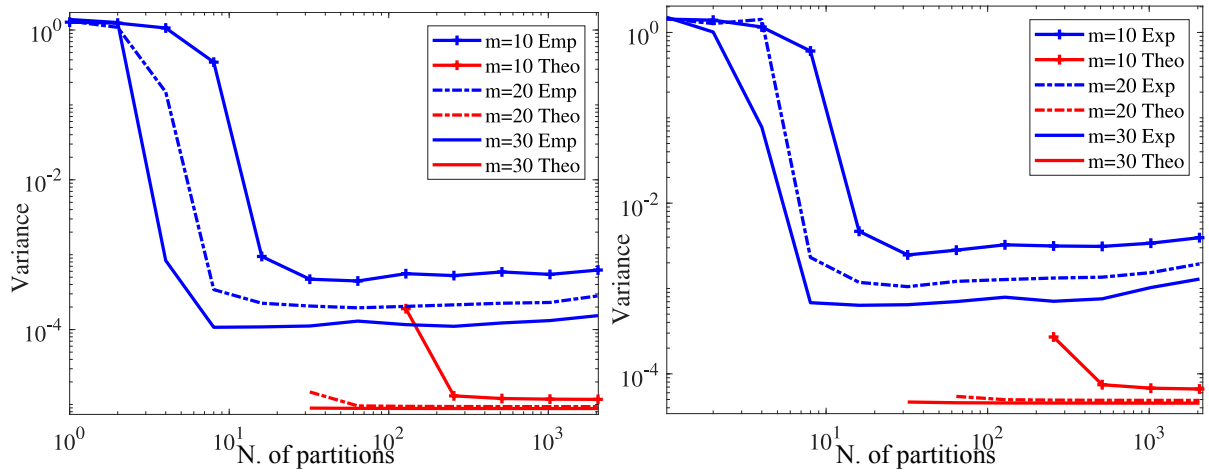


*Figure 23.* Comparison of the Cramer-rao lower bound and the empirical variance. Blue lines represent the empirical value, and red lines represent the theoretical value. Note that the red lines are shown only when the fisher information matrix is non-singular.

ios going from 6% to 30%. It can be seen that the values of $p$ presented in table 4 match those obtained in Fig. 23. Note that the red lines are shown only when the Fisher information matrix is non-singular, which, as expected, is close to the point of most minor empirical variance.

**3.5.4. Accuracy of the recovered covariance matrix.** The quality of the reconstructed covariance matrices was evaluated using the NMSE and the angle between the eigenvectors of the ground-truth covariance matrix and the recovered eigenvectors using Algorithm 3.1. In Fig. 24, the NMSE of the reconstructed covariance matrix using different types of matrices is

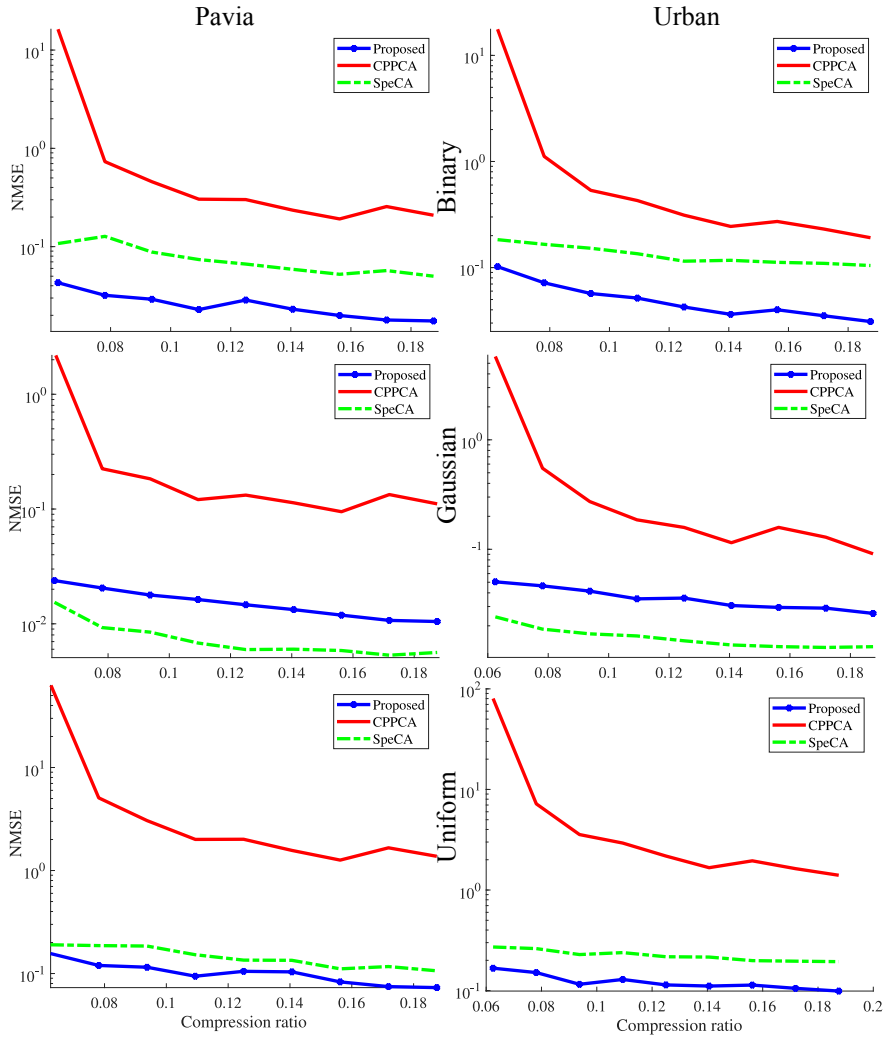


*Figure 24.* Average MSE of recovered covariance matrix when the compression ratio varies for the Urban image using the number of partitions given in table 4

shown. It can be seen that the proposed method outperforms both traditional methods (CPPCA, SpeCA), mainly when binary matrices are used. For the case of Gaussian matrices, **P** the proposed algorithm obtain comparable results to SpeCA.

Fig. 26 shows the angle gap obtained with the two different images for the three different types of random projections. Results in Fig. 26 are generated by running 20 times the proposed

algorithm, along with CPPCA and SpeCA. The angles of the recovered eigenvectors are averaged. The sensing protocol for the SpeCA algorithm is defined as $\mathbf{Y}_a = \mathbf{AX} \in \mathbb{R}^{m_a \times n}$, with $\mathbf{A} \in \mathbb{R}^{m_a \times l}$, $\mathbf{Y_b} = [\mathbf{B}_1\mathbf{x}_1, \mathbf{B}_2\mathbf{x}_2, \cdots, \mathbf{B}_n\mathbf{x}_n]$ and $\mathbf{B}_i \in \mathbb{R}^{m_b \times l}$, we set to $m_a = m - 1$ and $m_b = 1$. For these simulations, the signal was corrupted with additive Gaussian noise as in (50) to yield 20 dB of SNR. The results show that the angle gap of the recovered eigenvectors is less when the proposed algorithm is used with any type of projection matrix. Note that SpeCA produces similar results to the proposed algorithm when Gaussian projection matrices are used. However, the proposed algorithm outperforms SpeCA when Binary and Uniform matrices are used. Additionally, Figure 25 shows the running time for the three algorithms by varying the dimension of the subspace $m$. For the proposed method the stopping criterium was set to be the relative tolerance given by $\|\mathbf{\Sigma}_k - \mathbf{\Sigma}_{k-1}\|/\|\mathbf{\Sigma}_k\| \leq 1e^{-4}$. It can be seen that SpeCA requires 37 seconds for Pavia in contrast to 0.6 and 0.2 seconds for proposed and CPPCA, respectively. Even though CPPCA is the fastest method, the reconstruction quality is up to two orders of magnitude worst, as shown in Fig. 24. Note that the number of partitions for CPPCA and the proposed method is chosen following Lemma (2); when the dimension $m$ increases, the number of partitions $p$ decreases reducing the computation time.

**3.5.5. Error term and filtered gradient analysis.** In this section, the error term is numerically analyzed. For test purposes, we assume that the truth covariance is known so that the error matrices $\mathbf{R}_i$ are computed as $\mathbf{R}_i = \mathbf{S} - \mathbf{S}_i$, and the error is calculated as in (75). The covariance matrix is estimated using the proposed algorithm without filtering the gradient, and its eigenvectors are compared with the error term $\mathbf{B}$. This is because when no filtering is applied, we observe in the

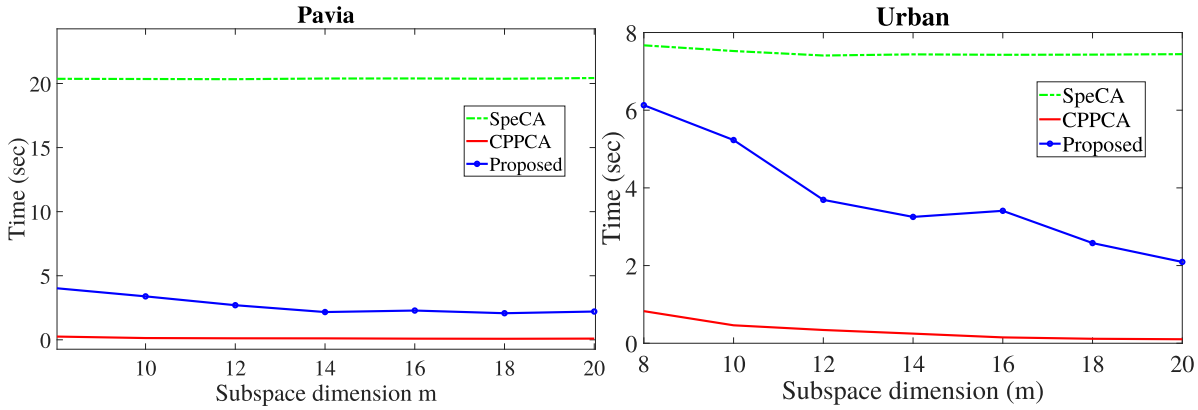


*Figure 25.* Comparison of the average execution time for SpeCA, CPPCA and the proposed algorithm. Note that the partition number is set to $p = l^2/m^2$ resulting in a reduction of the execution time when $m$ increases since the number of partitions decreases.

simulations that some eigenvectors are corrupted with high-frequency noise. Fig. 27 (left) shows the eigenvector's visual comparison when no filter is applied on the gradient and an eigenvector of the bias term (75). It can be seen that the fourth eigenvector of the recovered covariance matrix converges to the fourth eigenvector of $\mathbf{B}$, which computationally validates the statement in Lemma 1.

However, when the filtering procedure is applied using a Gaussian filter with $\sigma = 1$, this corrupted eigenvector converges to the actual one; this is shown in Fig. 27 (right). Further analysis is shown in Appendix 6.

### 3.6. Error term of the proposed gradient method

The gradient of $f(\mathbf{\Sigma})$ is given in (63). However, as mentioned above, the covariance matrices are not equal, thus (63) is rewritten as

$$\nabla \tilde{f}(\mathbf{\Sigma}) = \sum_{i=1}^{p} \mathbf{P}_i(\tilde{\mathbf{\Sigma}}_i - \mathbf{P}_i^T \mathbf{S}_i \mathbf{P}_i)\mathbf{P}_i^T + \tau \nabla \psi(\mathbf{\Sigma}). \tag{71}$$

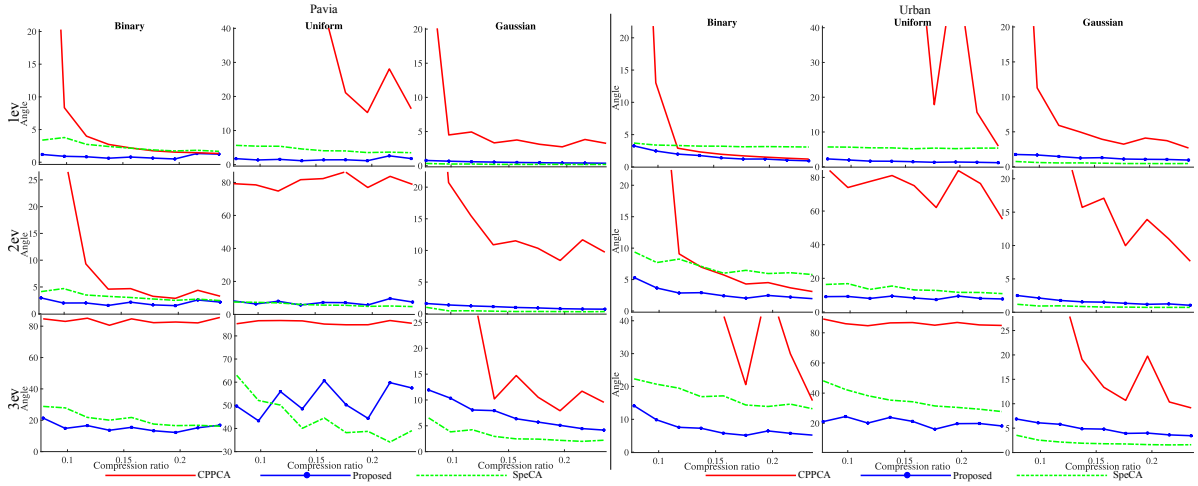


*Figure 26.* Average angle error of recovered eigenvectors with different compression ratios for Pavia (left), and Urban(right) images with different sensing matrices. Rows show the angle gap of the first, second, and third eigenvector.

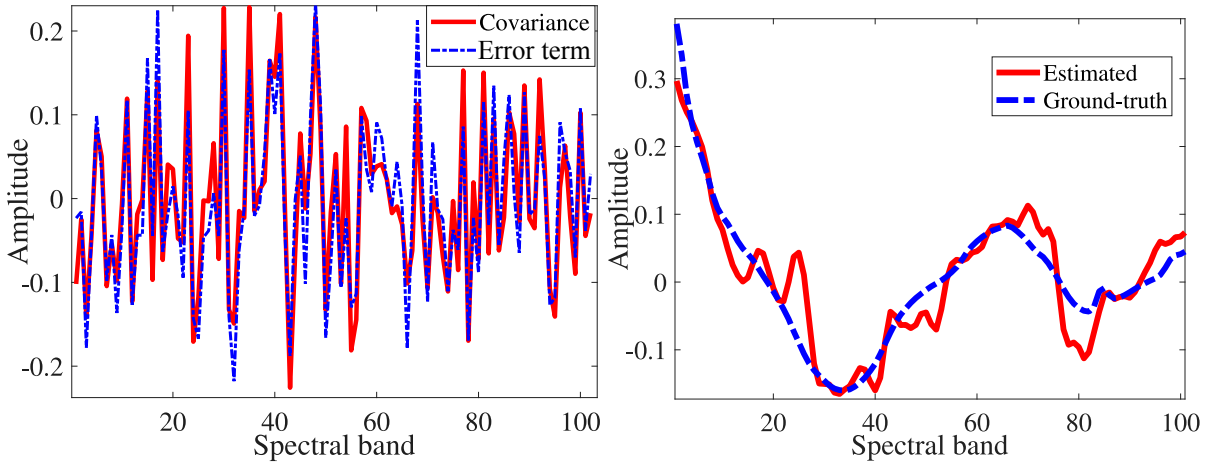


*Figure 27.* Comparison of the fourth eigenvector of the estimated covariance matrix with: (left) non-filtered gradient and the first eigenvector of the error term. (right) Filtered gradient and the fourth eigenvector of the truth covariance matrix.

Plugging (66) into (71) and assuming that $\mathbf{S} = \mathbf{\Sigma}$ we have that

$$\begin{aligned}\nabla \tilde{f}(\mathbf{\Sigma}) &= \sum_{i=1}^{p} \mathbf{P}_i(\tilde{\mathbf{\Sigma}}_i - \mathbf{P}_i^T(\mathbf{S}+\mathbf{R}_i)\mathbf{P}_i)\mathbf{P}_i^T + \tau\nabla\psi(\mathbf{\Sigma}) \\ &= \sum_{i=1}^{p} \mathbf{P}_i(\tilde{\mathbf{\Sigma}}_i - \mathbf{P}_i^T(\mathbf{\Sigma}+\mathbf{R}_i)\mathbf{P}_i)\mathbf{P}_i^T + \tau\nabla\psi(\mathbf{\Sigma}).\end{aligned} \tag{72}$$

After some algebraic operations, (72) can be rewritten as

$$\nabla \tilde{f}(\mathbf{\Sigma}) = \sum_{i=1}^{p} \mathbf{P}_i(\tilde{\mathbf{\Sigma}}_i - \mathbf{P}_i^T \mathbf{\Sigma} \mathbf{P}_i)\mathbf{P}_i^T - \sum_{i=1}^{p} \mathbf{P}_i \mathbf{P}_i^T \mathbf{R}_i \mathbf{P}_i \mathbf{P}_i^T + \tau \nabla \psi(\mathbf{\Sigma}). \tag{73}$$

Comparing (69) and (73), we can see that

$$\nabla \tilde{f}(\mathbf{\Sigma}) = \nabla f(\mathbf{\Sigma}) - \sum_{i=1}^{p} \mathbf{P}_i \mathbf{P}_i^T \mathbf{R}_i \mathbf{P}_i \mathbf{P}_i^T. \tag{74}$$

Hence, the error term of the gradient induced by the splitting procedure is

$$Error[\nabla \tilde{f}(\mathbf{\Sigma})] = -\sum_{i=1}^{p} \mathbf{P}_i \mathbf{P}_i^T \mathbf{R}_i \mathbf{P}_i \mathbf{P}_i^T. \tag{75}$$

**3.6.1. Image reconstruction.** The underlying signal is recovered with the estimated eigenvectors using the method described in (Fowler, 2009). In particular, given the matrix $\mathbf{W}_m \in \mathbb{R}^{l \times m}$ containing $m$ recovered eigenvectors, the signal is estimated as

$$\mathbf{X} = \mathbf{W}_m (\mathbf{P}^T \mathbf{W}_m)^{\dagger} \mathbf{Y}, \tag{76}$$

where † is the Moore-Penrose inverse. Using this approach, the image is reconstructed, and the performance is compared against SpeCA and CPPCA algorithms. Figure 28 shows the results for the Pavia centre image using the PSNR as a quality measurement.

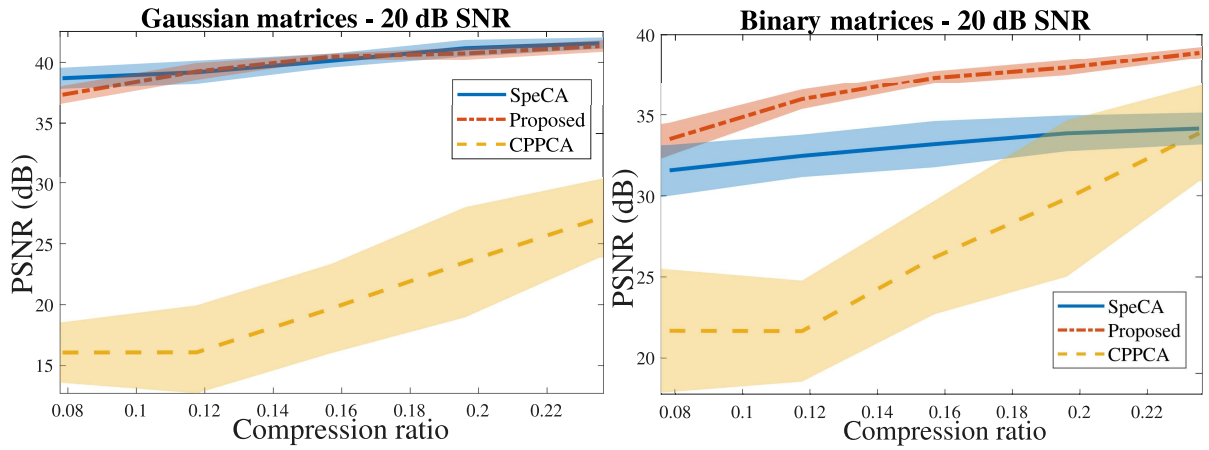


*Figure 28.* Comparison of the different reconstruction methods in terms of PSNR as a function of the compression ratio. The shaded areas represent the confidence interval.

It can be seen that using the estimator given in (76), the proposed method outperforms both state-of-art counterparts, specially CPPCA, which exhibits a large dispersion on the performance. Note that, SpeCA results are similar to those obtained when Gaussian matrices are used, but the proposed method outperforms by up to 5 dB using binary matrices.

**3.6.2. Optical implementation on DD-CASSI architecture.** Many implementable optical architectures model the sensing process as the vector formulation $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$(Correa et al., 2016b; Gehm et al., 2007b), where $\mathbf{x} = \text{vec}(\mathbf{X})$ and $\mathbf{H}$ is the sensing matrix. However, the proposed method requires dividing the sensing problem into multiple independent sub-problems and expressing them in matrix form. This partition can be achieved in architectures like DD-CASSI(Gehm et al., 2007b) or SSCSI(Lin et al., 2014b) since they preserve the spatial independence in the sensor, i.e., the codification/compression occurs only along the spectral dimension. To convert the vector problem into the multiples matrix sub-problems, note that the sensing problem

in DD-CASSI can be expressed as

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_n \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^T & 0 & \dots & 0 \\ 0 & \mathbf{P}_2^T & \dots & 0 \\ \vdots & \dots & \dots & \vdots \\ 0 & 0 & \dots & \mathbf{P}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix} + \mathbf{r}, \tag{77}$$

with $\mathbf{r}$ noise. From (77) it can be seen that each pixel is coded by a different sensing matrix $\mathbf{P}_i$. In fact, (77) is equivalent to (55) with $p = n$. Hence, if the number of sensing matrices is limited to $p < n$, (77) can be re-written as (Martín et al., 2015)

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_p \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^T & 0 & \dots & 0 \\ 0 & \mathbf{P}_2^T & \dots & 0 \\ \vdots & \dots & \dots & \vdots \\ 0 & 0 & \dots & \mathbf{P}_p^T \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_p \end{bmatrix} + \mathbf{E}, \tag{78}$$

where $\mathbf{X}_i \in \mathbb{R}^{l \times n/p}$ is a matrix whose columns are the pixels coded by the same matrix $\mathbf{P}_i$, and $\mathbf{E} = [\mathbf{N}_1^T, \dots, \mathbf{N}_p^T]^T$ is the noise. The schematic of the DD-CASSI, Figure 29, shows the distribution of the optical elements. The sensing process consists of four main steps: first, the scene goes through a prism that induces a dispersion effect; second, the scene is modulated by a binary coded aperture $\mathbf{C}$; third, a second prism undo the dispersion of the first prism, and fourth the scene is

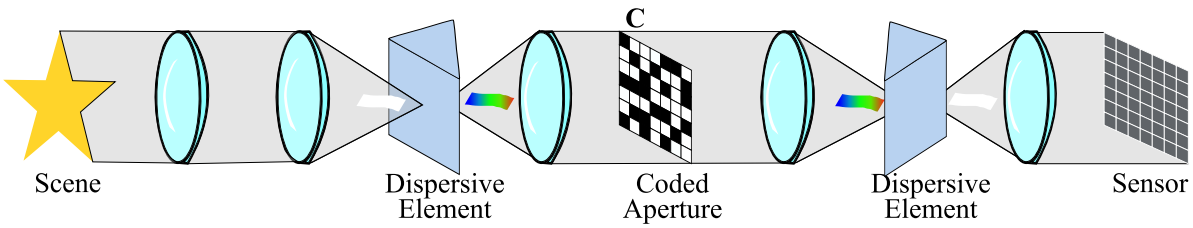


*Figure 29.* Schematic of DD-CASSI architecture.

integrated into the 2D sensor.

Limiting the number of sensing matrices $\mathbf{P}_i$ requires that the spatial distribution of the coded aperture $\mathbf{C}$ be designed to produce a limited number of code patterns in the spectral domain. One way to generate a limited number of sensing matrices consists of repeating a one-dimensional binary pattern $\mathbf{a} \in \mathbb{R}^{1\times p}$ along the spatial dimensions of the coded aperture $\mathbf{C}$. Specifically, the pattern $\mathbf{a}$ is repeated in each row of $\mathbf{C}$ in the same way; this concept is illustrated in Fig. 30. It can be seen that by repeating the pattern $\mathbf{a}$, we can construct the matrix $\mathbf{X}_1 = [\mathbf{x}_{(\mathbf{1},\mathbf{1})}, \mathbf{x}_{(1,5)}, \mathbf{x}_{(1,1)}, \mathbf{x}_{(1,1)}]$ since they share the same matrix $\mathbf{P}_1$.

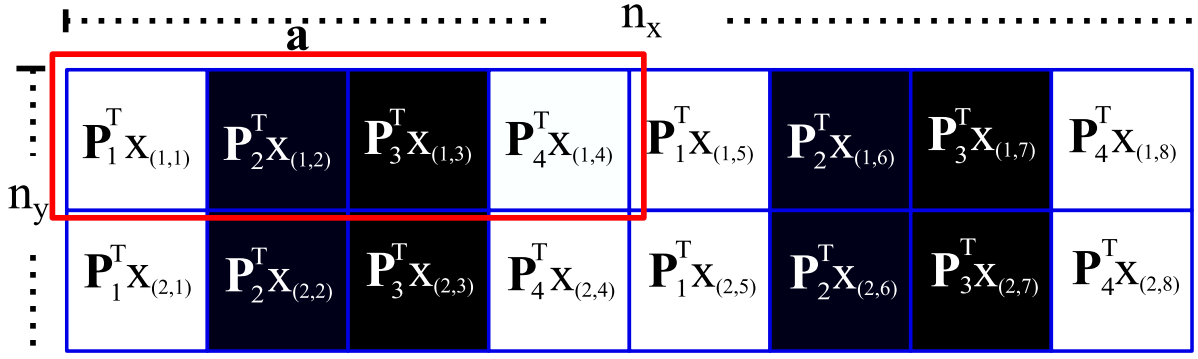


*Figure 30.* Sensing protocol design to limit the number of sensing matrices

In a multishot setup, the spatial distribution of the coded aperture must change i.e., there is a $\mathbf{C}_t$ for each snapshot $t$. Nevertheless, to preserve the subsets distribution, only the entries of $\mathbf{a}_t$

change at each instant $t$. Hence, the coded aperture spatial distribution follows

$$\text{vec}(\mathbf{C}_t) = \left[\mathbf{1}_{n_y} \otimes (\mathbf{1}_{N_x/p} \otimes \mathbf{a}_t)\right]. \tag{79}$$

We built a testbed in our laboratory as a proof-of-concept prototype based on(Marquez et al., 2021c, 2020c, 2021a); the optical setup is shown in Fig. 31. This optical device is made out of a Navitar lens (12$mm$ FixedFocal Length, MVL12M23 - 12$mm$ EFL, $f/1.4$) as the objective lens to image the scene onto the image plane of a matched achromatic doublet pair (Thorlabs MAP10100100-A, $f1 = 100.0mm, f2 = 100.0mm$) to propagate the incoming wavefront through a beam splitter until to a second matched achromatic doublet pair relay lens (Thorlabs MAP10100100-A, $f1 = 100.0mm, f2 = 100.0mm$). This second relay lens transmits the wavefront through a double Amici prism coupled to a rotation mount (Thorlabs CRM1P, 30$mm$ cage rotation mount, Ø1") to image a dispersed version of the scene onto the digital micromirror device (DMD, Texas Instruments, D4120). Taking advantage of the DMD's mirror surface, the now dispersed-modulated wavefront is returned through to the prism until the L2 lens, where the prism undoes the dispersion effect. The resulting dispersed-coded-dispersed wavefront propagates through BS until a third matched achromatic doublet pair relay lens (L3) (Thorlabs MAP105050-A, $f1 = 50.0mm, f2 = 50.0mm$). Finally, the L3 lens focuses the dispersed-coded-dispersed wavefront onto the sensor (Stingray F-080B, 4.65$\mu m$ pixel size).

The coded aperture was designed using (79) to produce a limited number of patterns with $m = 8$ snapshots. We placed the prism in a distance such that the dispersion generated $l = 37$

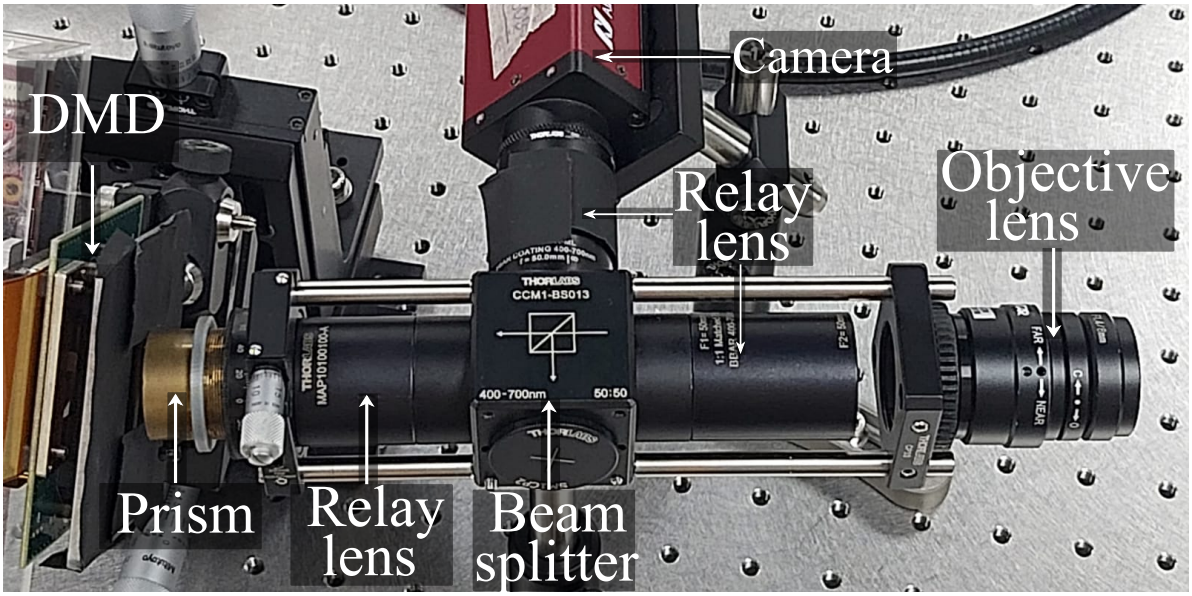


*Figure 31.* Optical implementation of the DD-CASSI architecture.

spectral bands in the sensor, and the spatial resolution of the scene was $356 \times 512$ pixels; this setup achieves a 79% of compression of the image. Based on Lemma (2), the optimal number of partitions must be $p > l^2/m^2 = 21.4$, so we generated 24 partitions. The sample mean was computed using (54) and subtracted from the measurements. Additionally, we used $\tau = 2e^{-4}$ and 200 iterations in the covariance recovery algorithm. We took the first five eigenvectors for the image reconstruction, and used them in (76). Overall, the whole process, including covariance matrix recovery and image reconstruction, took 0.85 seconds on average. Figure 32 shows an RGB composite of the hyperspectral image reconstructed and the RGB image captured with a commercial camera for comparison purposes. Fig. 32-b) shows four out of the 37 reconstructed spectral bands; these 37 spectral bands are in the range of 450 nm to 650 nm with a spectral resolution going from 2nm in blue spectral bands until 10nm in the red spectral bands. Figure 32-c) shows the recovered covariance matrix; Fig. 32-d) shows the sample mean and the three

eigenvectors associated with the largest eigenvalues. The figure shows that the RGB composite resembles the colors obtained with the commercial camera's high-resolution camera, which gives insights into the correct reconstruction.

### 3.7. Discussion

One limitation of the proposed method is that it requires multi-shot acquisition to correct covariance reconstruction. Specifically, in the reconstruction step, when a single shot is acquired, inverse problem (76) produces a rank-one solution that is not accurate. However, nowadays, cameras can acquire shots at a high-speed rate, reducing the impact of these limitations. Additionally, prior knowledge of the covariance matrix is required to set the convex set (e.g., Low-rank or Toeplitz). On the other hand, the partition of the data makes the method impractical to work with few realizations compared to the number of spectral bands. That is because the sample covariances matrices $\mathbf{S}_i$ will be poor estimators, and the error associated with the partition will increase dramatically. Nevertheless, the number of pixels is much greater than the number of spectral bands in imaging applications.

### 3.8. Conclusion

We proposed an algorithm to recover the covariance matrix from a set of compressive measurements using a strategy-based projection onto convex subsets. The algorithm is based on the projected gradient method. The theoretical results show that although the splitting procedure induces an error term, it can be mitigated using a filtered gradient. Additionally, this error is proportional to the number of partitions; nevertheless, more partitions improve the condition of the information matrix; thus, choosing the correct number of partitions is critical. For that reason, a

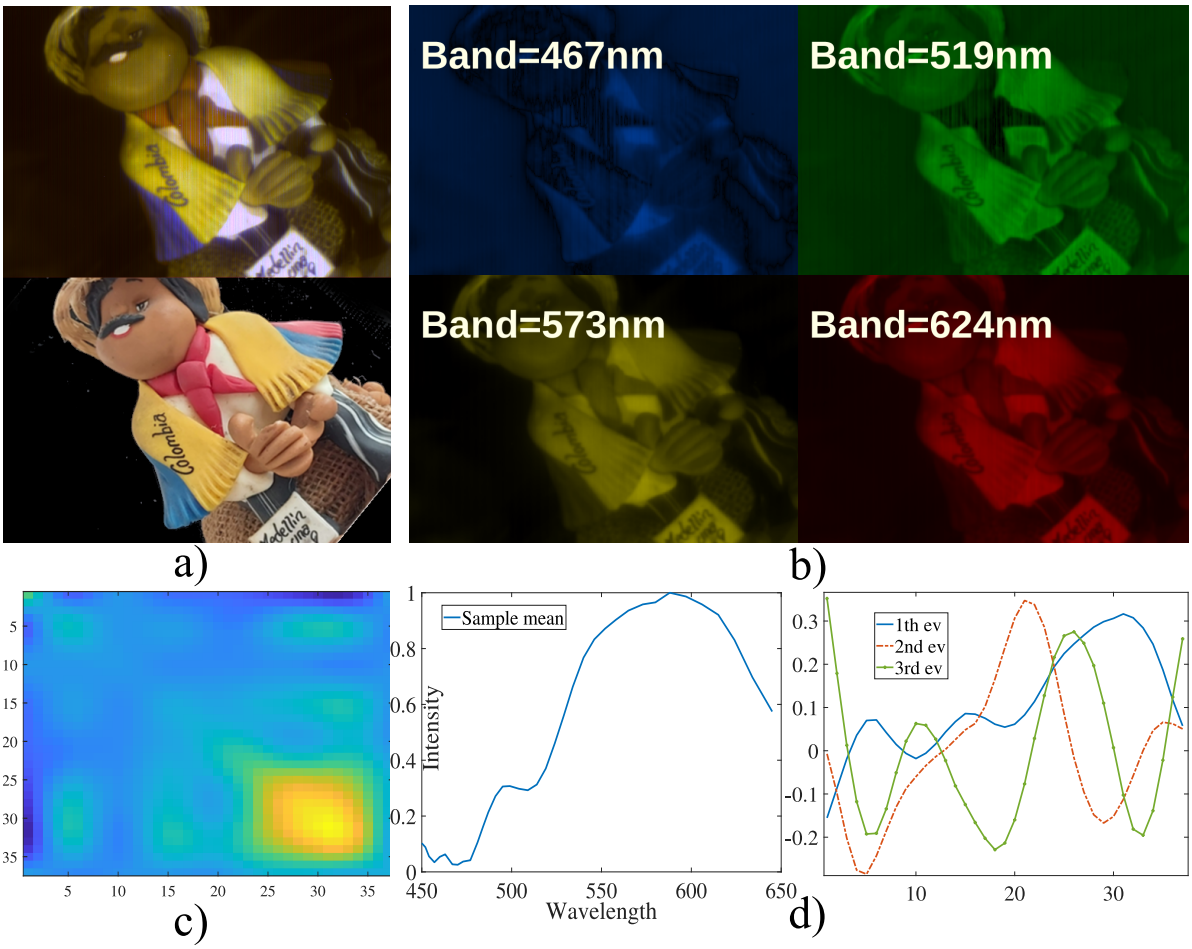


*Figure 32.* Reconstruction of the covariance matrix and hyperspectral image from data captured in our lab using the DD-CASSI optical architecture. a)-top RGB composite of the reconstructed hyperspectral image, a)-bottom RGB image acquired with a commercial camera. b) four out of the 37 spectral bands of the reconstructed hyperspectral image. c) Recovered covariance matrix. d) Sample mean and first three eigenvectors of the covariance matrix.

lower bound for the optimal number of partitions is proposed. Experimental results show that the proposed method outperforms state-of-art algorithms CPPCA and SpeCA. The experiments were performed using two different hyperspectral images, for which the proposed method attained better results in terms of MSE and angle GAP, which translates in a gain of up to 10 dB of PSNR in comparison with CPPCA and up to 4 dB PSNR concerning SpeCA. Additionally, the algorithm was tested with real data from the laboratory using DD-CASSI architecture. It can be seen that the reconstruction process is fast and robust since the RGB composite resembles the RGB image of the scene.

# 4. Covariance Estimation for Spectral Video Recovery using a Projected Gradient based Algorithm

*This chapter addresses the second and third objective of the thesis:*

- *To design an algorithm based on the gradient descent method to recover the first and second sample statistical moments from low-dimensional random projections.*
- *To test the performance of the proposed algorithm to recover the sample statistics in hyperspectral imaging reconstruction*

## 4.1. Introduction

Chapter 3 introduced a novel algorithm and an optical architecture to recover the covariance matrix. However, the proposed methodology requires multiple acquisitions at different times to recover an accurate CM. this approach relegates the temporal CM estimation to a sequential uncorrelated problem, i.e., ignoring the inter-temporal CM correlation. Therefore, it is desirable to design an SCSV methodology based on CCS theory to estimate the CM exploiting the inter-temporal correlation from a single compressive measurement per frame. This single snapshot approach enables to use of the CCS theory for spectral video applications.

Spectral video (SV) can be seen as a 4D tensor that contains the spatial $(x,y)$, spectral $(\lambda)$, and temporal $(t)$ information of a dynamic scene. SV plays a central role in high-level computer vision applications such as classification (Hu and Hsu, 2013; Leon-Lopez et al., 2022), detection (Liu et al., 2019), food safety (Gao et al., 2020; Al-Sarayreh et al., 2020), among others. Tra-

ditional SV acquisition relies on scanning approaches that compromise the temporal resolution, i.e., the changes in the scene must be slower than the acquisition of the frame. In contrast, snapshot compressive spectral video (SCSV) acquires 2D projections of the 4D tensor, preserving the spatio-spectral features with high accuracy and reducing the acquisition times.

Conventional SCSV optical systems use spatial light modulators and dispersive elements to encode and compress the 4D tensor into a set of 2D measurements. One of the most used methods in the SCSV state-of-the-art is the binary mask-based method. Their increasing popularity is associated with the high codification variability, low calibration complexity, high light throughput, and high-speed modulation rates(Arguello and Arce, 2012b; Correa et al., 2016a; Marquez et al., 2021b, 2020a; Correa et al., 2016c). Specifically, this sensing geometry is inspired by the compressive coded aperture spectral imaging (CASSI) architecture. CASSI-based imagers fall on two sensing approaches, multiple acquisitions (Arguello and Arce, 2014) or single acquisition per frame (Gehm et al., 2007b). The first one enables better reconstruction performances than the single acquisition approach, but it is unfeasible in scenarios with fast temporal changes. In contrast, the single acquisition approach enables the acquisition of fast-moving targets at the expense of spatial-spectral reconstruction quality. Although both methods enable the acquisition and recovery of SV at different frame rates, their main shortcoming lies in using high-complexity reconstruction algorithms that limit their feasibility in real-time computer vision applications(Arguello et al., 2013; Zhang et al., 2019).

This chapter proposes a snapshot spectral video imager founded on the CCS theory, named compressive covariance spectral video (CoCoS-Vi). Specifically, the CoCoS-Vi sensing geometry

aims to generate multi-views of the scene with a lenslet array followed by a disperse-code-disperse process using a single dispersive element, a digital mirror device (DMD), and a beam splitter. The resulting compressed measurement contains multiple low-spatial resolutions encoded versions per frame. These encoded views can be seen as a different snapshot per frame, e.i., a virtual multishot approach. Moreover, we propose a reconstruction algorithm based on a spectro-temporal low-rank approximation of the CM. This reconstruction methodology requires a specific coded aperture design to re-cast the traditional vector sensing model into a tensor sensing model. CoCoS-Vi claims are validated via experimental measurements acquired with a proof-of-concept prototype. An spectral scene was acquired and reconstructed using the CoCoS-Vi methodology achieving a reconstruction time of 0.12 seconds per frame for a spectral image of dimension $151 \times 196 \times 15$.

### 4.2. Spectro-temporal low-rank covariance matrix estimation

**4.2.1. Discrete sensing model.** A hyperspectral video can be represented as a 4D tensor $\underline{\mathbf{F}} \in \mathbb{R}^{N_x \times N_y \times N_\lambda \times N_t}$, where $N_x \times N_y$ indexes the spatial dimension, $N_\lambda$ the wavelength bands, and $N_t$ the frames. The CoCos-Vi sensing protocol is based on the DD-CASSI multishot sensing model, which can expressed as

$$\begin{bmatrix} \mathbf{g}_{i1} \\ \mathbf{g}_{i2} \\ \vdots \\ \mathbf{g}_{in} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1^T & 0 & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2^T & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{H}_n^T \end{bmatrix} \begin{bmatrix} \mathbf{f}_{i1} \\ \mathbf{f}_{i2} \\ \vdots \\ \mathbf{f}_{in} \end{bmatrix} + \boldsymbol{\varepsilon}_i, \tag{80}$$

where $n = N_x N_y$, $\mathbf{f}_{ij} \in \mathbb{R}^{N_\lambda}$ and $\mathbf{g}_{ij} \in \mathbb{R}^K$ are the spectral signature and compressed measurement for the $j$-th pixel in the $i$-th frame, respectively, with $j = \{1,\ldots,n\}$ and $i = \{1,\cdots,N_t\}$; $K \in \mathbb{N}$ is the total of snapshots per-frame, $\mathbf{H}_j \in \mathbb{R}^{N_\lambda \times K}$ represents the sensing matrix for the $j$-th spectral signature, and $\boldsymbol{\varepsilon}_i \in \mathbb{R}^{nK}$ is the additive noise. Based on (Monsalve et al., 2021, 2022b), the number of sensing matrix in (80) can be limited to a small set of $\rho \in \mathbb{N}$ primary sensing matrices $\mathbf{P}_\ell \in \mathbb{R}^{N_\lambda \times K}$ (with $\ell = \{1,\cdots,\rho\}$ and $\rho < n$) that repeats arbitrarily through the block-diagonal matrix. Taking this into account, the sensing model in (80) can be re-cast in matrix form as

$$\begin{bmatrix} \mathbf{G}_{i1} \\ \mathbf{G}_{i2} \\ \vdots \\ \mathbf{G}_{i\rho} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_2^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & \mathbf{P}_\rho^T \end{bmatrix} \begin{bmatrix} \mathbf{F}_{i1} \\ \mathbf{F}_{i2} \\ \vdots \\ \mathbf{F}_{i\rho} \end{bmatrix} + \mathbf{E}_i \,, \tag{81}$$

where $\mathbf{F}_{i\ell} \in \mathbb{R}^{N_\lambda \times (n/\rho)}$ contains all the spectral signatures $\mathbf{f}_{ij}$ that share the same primary sensing matrix $\mathbf{P}_\ell$ for the $i$-th frame, $\mathbf{G}_{i\ell} \in \mathbb{R}^{K \times (n/\rho)}$ is the matrix version of the compressed measurements, and $\mathbf{E}_i \in \mathbb{R}^{(\rho \cdot K) \times n/\rho}$ is the sensor noise. CoCoS-Vi sensing protocol aims to avoid the multiple snapshots dependence using a lenslet array to acquire different encode versions simultaneously at the cost of the spatial resolution. Specifically, each row of $\mathbf{G}_{i\ell}$ is the same arbitrary spatial position replicated by the lenslet array but modulated by a different section of the coded aperture. Hence, from here and onward on the document, the variable $K$ refers to the number of lenses that compose

the lenslet array.

**4.2.2. Covariance matrix recovery.** The objective of the CoCos-Vi methodology is to recover the spectral video via the spectral covariance matrix $\blacksquare_i \in \mathbb{R}^{N_\lambda \times N_\lambda} = \mathbb{E}_\ell[\mathbf{f}_{i\ell}\mathbf{f}_{i\ell}^T]$ using the sample covariance matrices $\tilde{\mathbf{S}}_{i\ell} = \mathbf{G}_{i\ell}\mathbf{G}_{i\ell}^T/(n/\rho) \in \mathbb{R}^{K\times K}$. The state-of-the-art CM approaches aim to solve an uncorrelated temporal problem given by(Monsalve et al., 2021, 2022b; Blanco et al., 2021):

$$\mathbf{\Sigma}_i^* = \underset{\mathbf{\Sigma}_i \in S_{++}}{\arg\min} \quad \sum_{\ell=1}^{\rho} \|\tilde{\mathbf{S}}_{i\ell} - \mathbf{P}_\ell^T \mathbf{\Sigma}_i \mathbf{P}_\ell\|_F^2 + \tau \mathrm{Tr}(\mathbf{\Sigma}_i), \tag{82}$$

where Tr represents the matrix trace, $\tau$ a regularizer parameter that penalize the solution rank, and $S_{++}$ is the positive-semidefinitive set. Note that problem (82) lacks exploiting the temporal correlation. Therefore, to exploit the intrinsic spectral-temporal correlation in the natural videos and the resulting DMD time-invariant structure in the CoCoS-Vi sensing protocol, we propose reconstructing the third-order tensor $\underline{\mathbf{\Sigma}} \in \mathbb{R}^{N_\lambda \times N_\lambda \times N_t}$ at once. Let's $\mathbf{\Sigma}_i \in \mathbb{R}^{N_\lambda \times N_\lambda}$ be the $i$-th slice of the tensor $\underline{\mathbf{\Sigma}}$. Then, the recovery problem can be defined as

$$\begin{aligned} \underline{\mathbf{\Sigma}}^* = & \arg\min_{\underline{\mathbf{\Sigma}}} \sum_{i=1}^{N_t} \sum_{\ell=1}^{\rho} ||\tilde{\mathbf{S}}_{i\ell} - \mathbf{P}_\ell^T \mathbf{\Sigma}_i \mathbf{P}_\ell||_F^2 \\ & + \tau \sum_{i=1}^{N_t} \mathrm{Tr}(\mathbf{\Sigma}_i) \\ & \text{s.t. } \mathrm{rank}(R_3(\underline{\mathbf{\Sigma}})) = r \text{ and } \mathbf{\Sigma}_i \in S_{++} \forall i, \end{aligned} \tag{83}$$

where $R_3(\cdot) : \mathbb{R}^{N_\lambda \times N_\lambda \times N_t} \to \mathbb{R}^{N_t \times (N_\lambda \cdot N_\lambda)}$ is a function that takes a third-order tensor as input and outputs a two-order tensor, commonly know as the unfolding 3-mode (Rabanser et al., 2017). Note

that forcing low-rank approximations in the unfolding 3-mode of the tensor promotes reconstructions with high temporal correlation. As (83) is non-convex due to the rank restriction, it can be rewritten in a convex unconstrained form given by

$$\begin{aligned}\underline{\mathbf{\Sigma}}^* =& \arg\min_{\underline{\mathbf{\Sigma}}} \sum_{i=1}^{N_t}\sum_{\ell=1}^{\rho} \|\tilde{\mathbf{S}}_{i\ell} - \mathbf{P}_\ell^T \mathbf{\Sigma}_i \mathbf{P}_\ell\|_F^2 \\ &+ \tau \sum_{i=1}^{N_t} \mathrm{Tr}(\mathbf{\Sigma}_i) + \mu \|R_3(\underline{\mathbf{\Sigma}})\|_* + \sum_{i=1}^{N_t} \iota_S(\mathbf{\Sigma}_i),\end{aligned} \tag{84}$$

where $\iota_S$ is the indicator function for the set of positive semidefinite matrices, $\mu$ is a regularization parameter, and $\|.\|_*$ is the nuclear norm. Problem (84) can be seen as the sum of a differentiable convex term

$$f(\underline{\mathbf{\Sigma}}) = \sum_{i=1}^{N_t}\sum_{\ell=1}^{\rho} \|\tilde{\mathbf{S}}_{i\ell} - \mathbf{P}_\ell^T \mathbf{\Sigma}_i \mathbf{P}_\ell\|_F^2 + \tau \sum_{i=1}^{N_t} \mathrm{Tr}(\mathbf{\Sigma}_i), \tag{85}$$

and a convex non differentiable term

$$g(\underline{\mathbf{\Sigma}}) = \mu \|R_3(\underline{\mathbf{\Sigma}})\|_* + \sum_{i=1}^{N_t} \iota_S(\mathbf{\Sigma}_i). \tag{86}$$

Problem (84) is solved using a proximal gradient method, that implies estimated the gradient of (85) and the proximal operator of (86). Specifically, the gradient and the proximal operator are defined as

$$\nabla f(\underline{\mathbf{\Sigma}}) = \sum_{i=1}^{N_t}\sum_{\ell=1}^{\rho} \mathbf{P}_\ell (\tilde{\mathbf{S}}_{i\ell} - \mathbf{P}_\ell^T \mathbf{\Sigma}_i \mathbf{P}_\ell) \mathbf{P}_\ell^T + \tau_N \mathbf{I}, \tag{87}$$

and

$$P_g(\underline{\mathbf{\Sigma}}) = (\underline{\mathbf{\Sigma}}_* + \underline{\mathbf{\Sigma}}_S)/2. \tag{88}$$

where $\tau_N = N_t \times \tau$, $\lambda_k$ is the learning rate, $\underline{\mathbf{\Sigma}}_*$ and $\underline{\mathbf{\Sigma}}_S$ are the nuclear norm and the positive-semidefinite proximal operator outputs, respectively. Specifically, these two operators can be defined as $\underline{\mathbf{\Sigma}}_* = P_*(R_3(\underline{\mathbf{\Sigma}}))$ and $\underline{\mathbf{\Sigma}}_S = P_S(\{\mathbf{\Sigma}_i\}_1^{N_t}))$, where

$$P_*(\mathbf{X}) = \mathbf{U}\mathrm{s}(\mathbf{\Lambda})\mathbf{V}^T, \tag{89}$$

and

$$P_S(\mathbf{X}) = \mathbf{W}(\tilde{\mathbf{\Lambda}})_+\mathbf{W}^T, \tag{90}$$

with $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ is the singular value decomposition, $\mathrm{s}(.)$ is the soft threshold operator (Cai et al., 2010; Gelvez and Arguello, 2021), $\mathbf{X} = \mathbf{W}\tilde{\mathbf{\Lambda}}\mathbf{W}^T$ is the eigenvalue decomposition and $(\tilde{\mathbf{\Lambda}})_+$ is the nonnegative part of $\mathbf{\Lambda}$ (Bioucas-Dias et al., 2014). Note that the proximal $P_g$ can be expressed as the average of the individual proximal (89) and (90) because $g(\dot{)}$ is the average of the two convex terms (Parikh and Boyd, 2013). Finally, the update step for the proximal gradient method is given by (Beck, 2017)

$$\underline{\mathbf{\Sigma}}^{k+1} = P_g(\underline{\mathbf{\Sigma}}^k - \lambda_k \nabla f(\underline{\mathbf{\Sigma}}^k)) \tag{91}$$

The proximal gradient algorithm is compiled as follows

**Algorithm 4.1** Proximal gradient algorithm

1: $\mathbf{\Sigma}^0, \tau, \lambda_0$

2: **while** stopping criteria is not satisfied **do**

3: pick $\lambda_k > 0$ ▷ Armijo search

4: $\underline{\mathbf{\Sigma}}^{k+1} \leftarrow \underline{\mathbf{\Sigma}}^k - \lambda_k \nabla f(\underline{\mathbf{\Sigma}}^k)$

5: $\underline{\mathbf{\Sigma}}^{k+1} \leftarrow P_g(\underline{\mathbf{\Sigma}}^{k+1})$

6: **end while**

Algorithm 4.1 is summarized in four main steps. First, initializations for the covariance matrices $\mathbf{\Sigma}$, regularizer $\tau$, learning rate $\lambda$ are set. In line 3, the learning rate $\lambda_k$ is updated based on the Armijo search(Iusem, 2003), then, in line 4, the covariance matrix is updated following the gradient. Finally, the result is updated using the proximal operator $P_g$.

The reconstruction of the spectral video is accomplished by pseudoinverse regularized by the eigenvectors. This step is modeled as

$$\tilde{\mathbf{F}}_{ij} = \mathbf{W}_i(\mathbf{P}_j^T \mathbf{W}_i)^\dagger \mathbf{G}_{ij}, \tag{92}$$

where $\mathbf{W}_i$ are the eigenvectors associated with the greatest eigenvalues of the covariance matrix $\mathbf{\Sigma}_i$.

### 4.3. Simulations and Experimental validations

**4.3.1. Simulations.** Simulations used a spectral video of dimensions $512 \times 480 \times 31$ and 20 frames and the acquisition emulates a $2 \times 2$ lenslet. The coded aperture spatial dis-

tribution is generated following the process described in (Monsalve et al., 2022b) from a random Gaussian pattern which generated 64 partitions $\rho$. The covariance matrix is reconstructed using the proposed Algorithm 4.1 with hyperparameters $\tau = 0.001$ and 100 iterations. We compare against the results obtained by using the uncorrelated covariance temporal algorithm proposed in (Monsalve et al., 2022b)(termed as CoCoS-Vi w/o temporal) and a PnP ADMM-based algorithm that reconstructs the video without computing the covariance matrix(Yuan et al., 2020); for that, the CM for the PnP algorithm is computed directly from the recovered video. Fig. 33 illustrates five reconstructed CM for frames {1, 10, 15, and 20} for the proposed algorithm and the two state-of-the-art algorithms. Each covariance matrix image contains the normalized mean squared error as a title for numerical comparison. Note that the proposed algorithm achieves an NMSE around one and a half order of magnitude less than the PnP-ADMM-based algorithm and almost half of the obtained with the CoCoS-Vi w/o temporal.

The spectral video is also reconstructed and the performance is evaluated using Peak Signal-to-Noise Ratio (PSNR), structural similarity index measure (SSIM) and Spectral Angle Mapper(SAM) metrics. Fig. 34 shows the box plot for the 3 metrics. Note that the proposed method outperforms both state-of-the-art algorithms in average for all the metrics. The low variance of the results in the PnP-ADMM-based algorithm can be the result of the over-smoothing applied to the statistical behavior shown in the covariance matrix estimation. Additionally, the proposed method requires around 0.43 seconds to reconstruct each frame in contrast to 163 seconds of the PnP-ADMM-based method, which is a speedup of 379x.

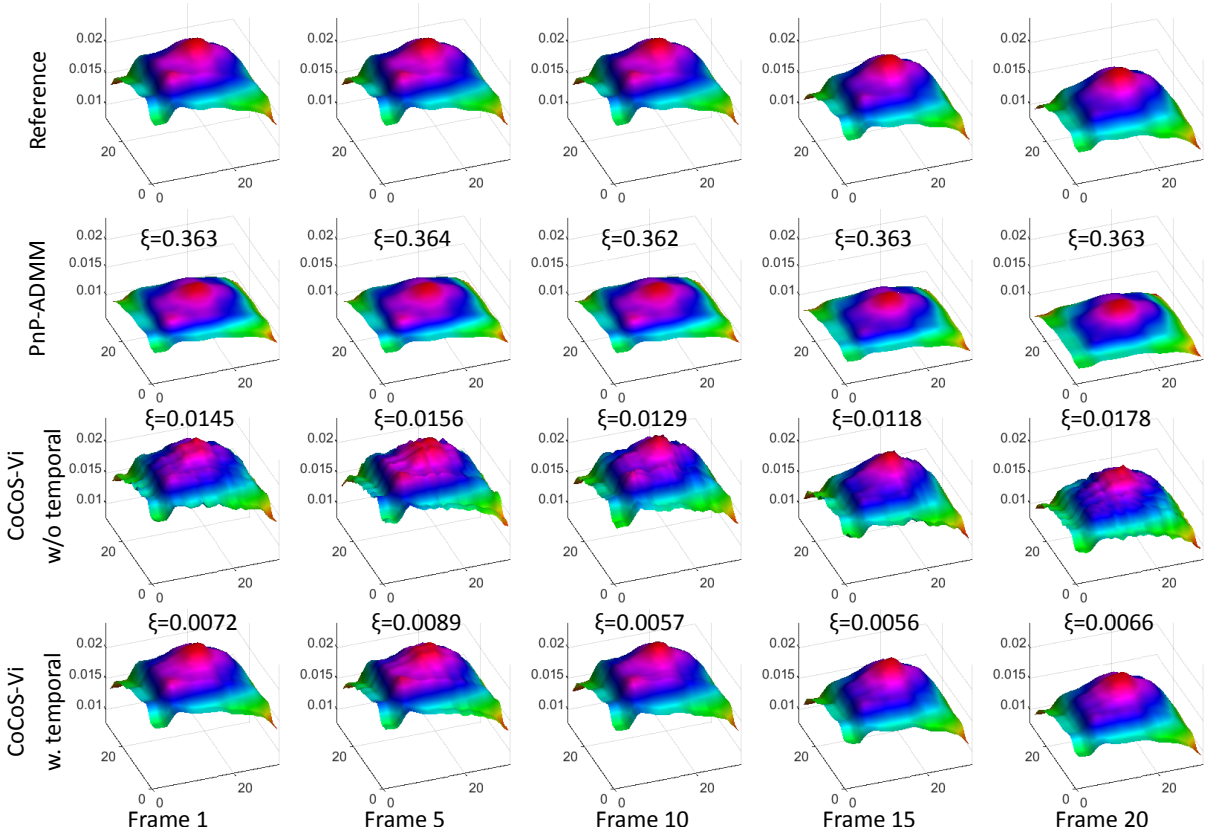


*Figure 33.* Five selected recovered spectral covariance matrices for the frames {1,5,10,15,20}. Here, for the PnP-ADMM approach, the covariance matrix is calculated directly from the reconstructed datacube. The NMSE metric is used to measure the CM reconstruction, defined as $\xi = \|\mathbf{\Sigma}_i - \tilde{\mathbf{\Sigma}}_i\|_F / \|\mathbf{\Sigma}_i\|_F$.

**4.3.2. CoCoS-Vi testbed.** We built a testbed in our laboratory to demonstrate the validity of the proposed ideas, through a proof-of-concept prototype, as shown in Fig. 35. This prototype uses a matched achromatic doublet pair relay lens (Lens 4) (Thorlabs MAP10100100-A, $f1 = 100.0mm, f2 = 100.0mm$) as an objective lens to propagate the incoming wavefront through a lenslet array ($38.1mm$ focal length, Edmunds 63-230, $46 \times 46mm$ Lenslet Array, $4 \times 3mm$ Lenslets). Then the lenslet array and the doublet lens located in tandem generate multi-views of

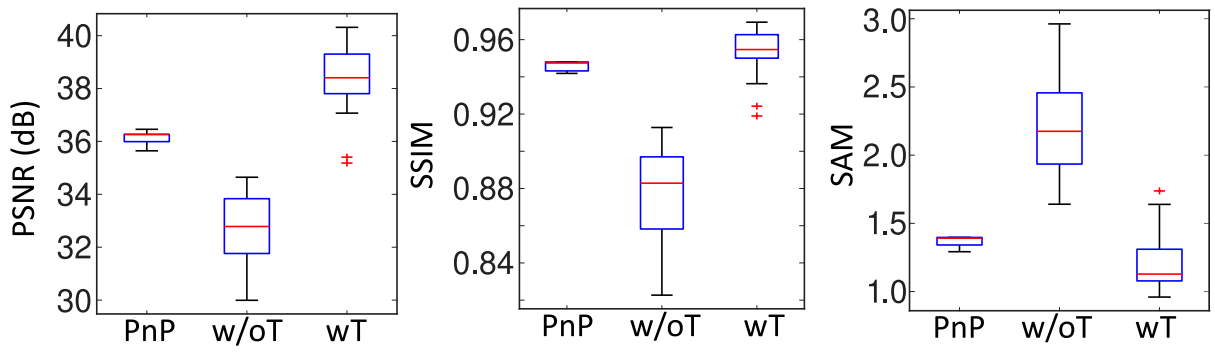


*Figure 34.* Numerical comparison of the spectral video reconstruction quality; PnP represents the results for the PnP-ADMM based algorithm, Cw/oT represents the results for the algorithm proposed in (Monsalve et al., 2022b) and CT is the proposed CoCos-Vi methodology.

the scene to image it by a Navitar lens (Lens 1) (12$mm$ Fixed Focal Length, MVL12M23 - 12$mm$ EFL, $f/1.4$) to their back focal length. Following, a matched achromatic doublet pair lens (Lens 2) (Thorlabs MAP105050-A, $f1 = 50.0mm, f2 = 50.0mm$) propagates and images a magnificate multi-view scene version onto the image plane of a matched achromatic doublet pair lens (Lens 3) (Thorlabs MAP105050-A, $f1 = 50.0mm, f2 = 50.0mm$) to propagate the incoming wavefront through a beam splitter until to a matched achromatic doublet pair relay lens (Lens 4) (Thorlabs MAP10100100-A, $f1 = 100.0mm, f2 = 100.0mm$). Lens 4 transmits the wavefront through a double Amici prism coupled to a rotation mount (Thorlabs CRM1P, 30$mm$ cage rotation mount, Ø1") to image a dispersed version of the scene onto the digital micromirror device (DMD, Texas Instruments, D4120). Taking advantage of the DMD's mirror surface, the now dispersed-modulated wavefront is returned through to the prism until lens 4, where the prism undoes the dispersion effect. The resulting dispersed-coded-dispersed wavefront propagates through BS until a mounted achromatic doublets lens (Lens 5) (100$mm$ Focal length, Thorlabs AC254-100-A-ML - ARC: 400-

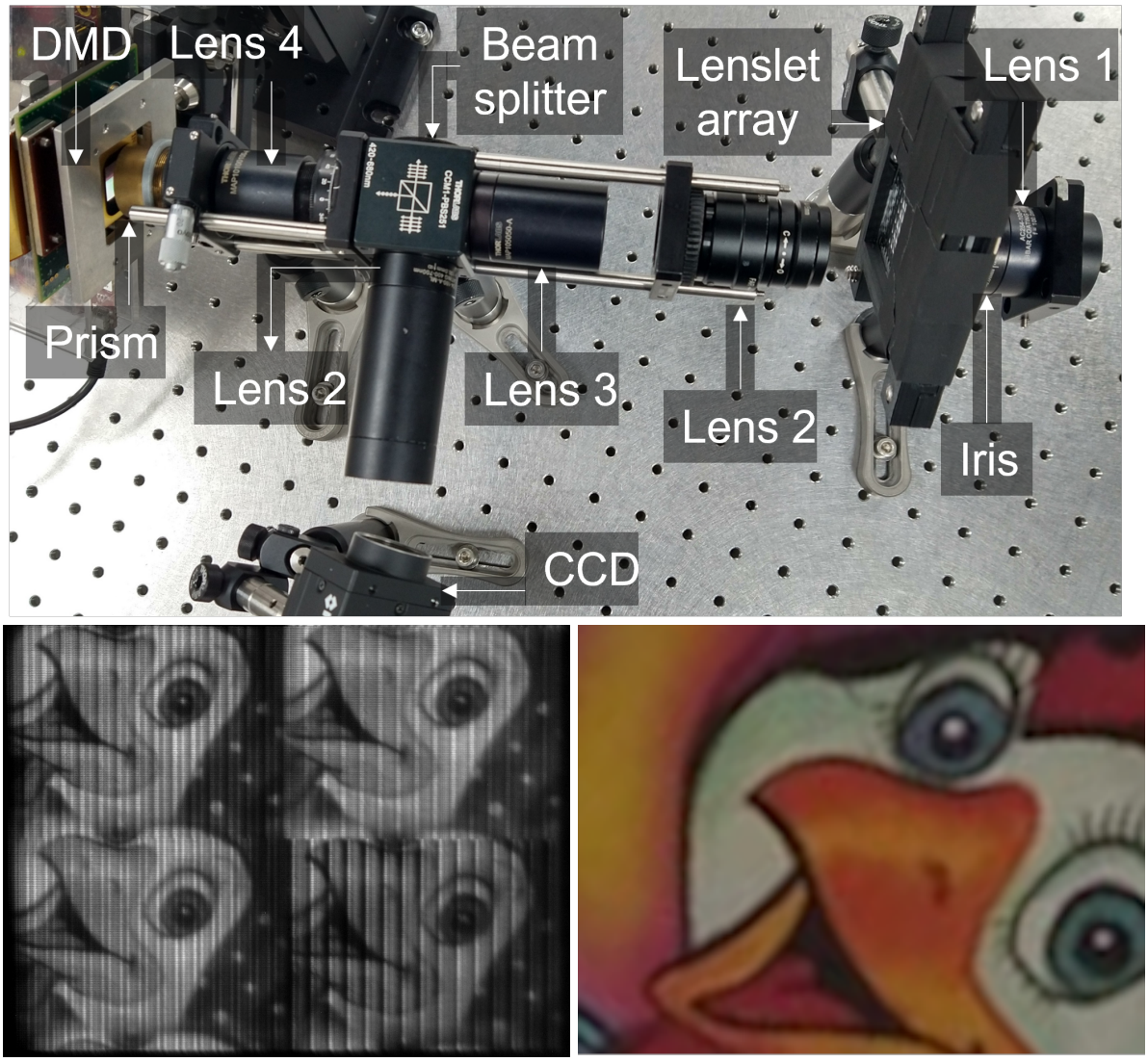


*Figure 35.* (Upper row) CoCoS-Vi optical setup. (Bottom row, left column) CoCoS-Vi compressed raw measurements for Chicken video in the first frame and (Bottom row, right column) RGB reference acquired with a commercial camera.

700 nm). Finally, this lens focuses the dispersed-coded-dispersed wavefront onto the sensor.

**4.3.3. Experimental results.** The experiments evaluate one target scene named Chicken, for which and array of $2 \times 2$ lenslet were used. The raw compressive projections exhibit

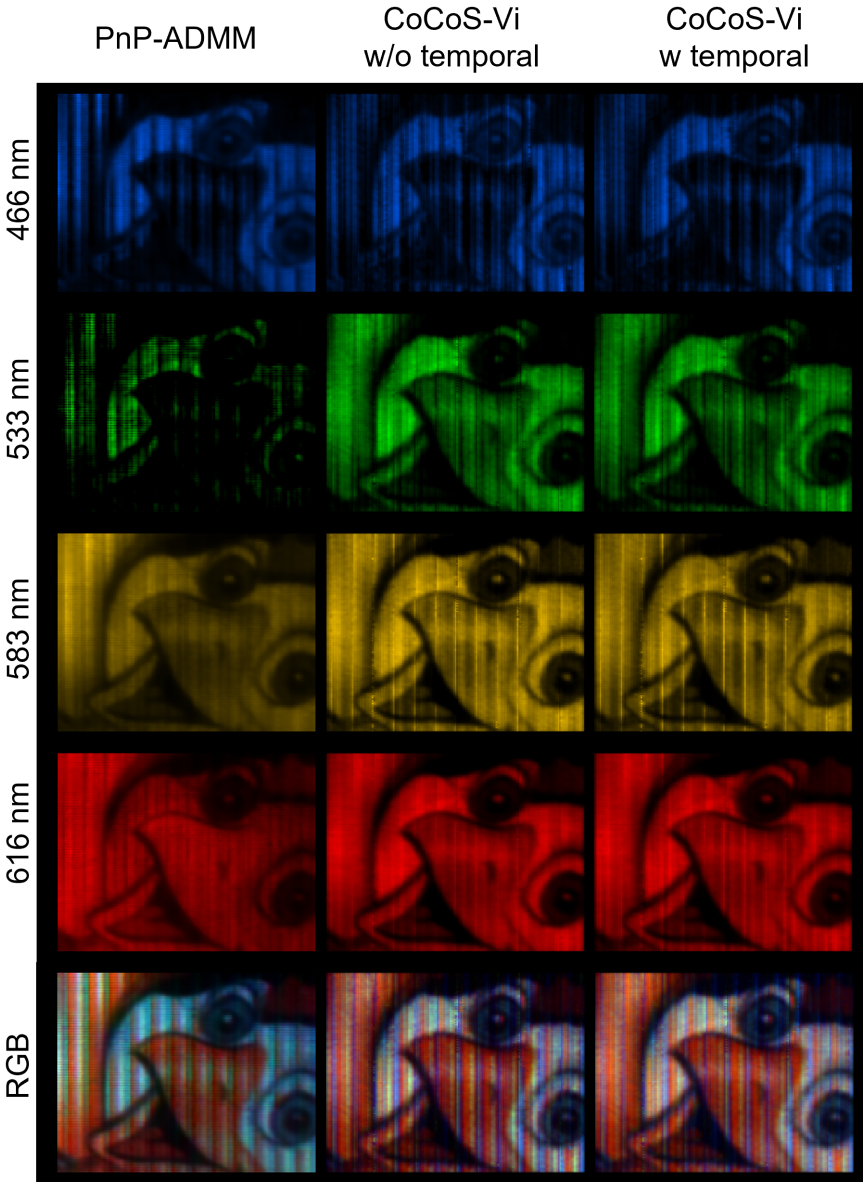


*Figure 36.* Four selected spectral bands and the RGB composite from the reconstruction

.

a spatial resolution of $151 \times 196$ pixels with $N_\lambda = 15$ spectral bands ranging from 490 to 650 *nm* and a total of 20 frames. The coded aperture spatial distribution is generated following the process described in (Monsalve et al., 2022b) from a random binary pattern which generated 16 partitions $\rho$. The raw measurements for the frame 1 and 20 are shown in Fig. 35(bottom row)).

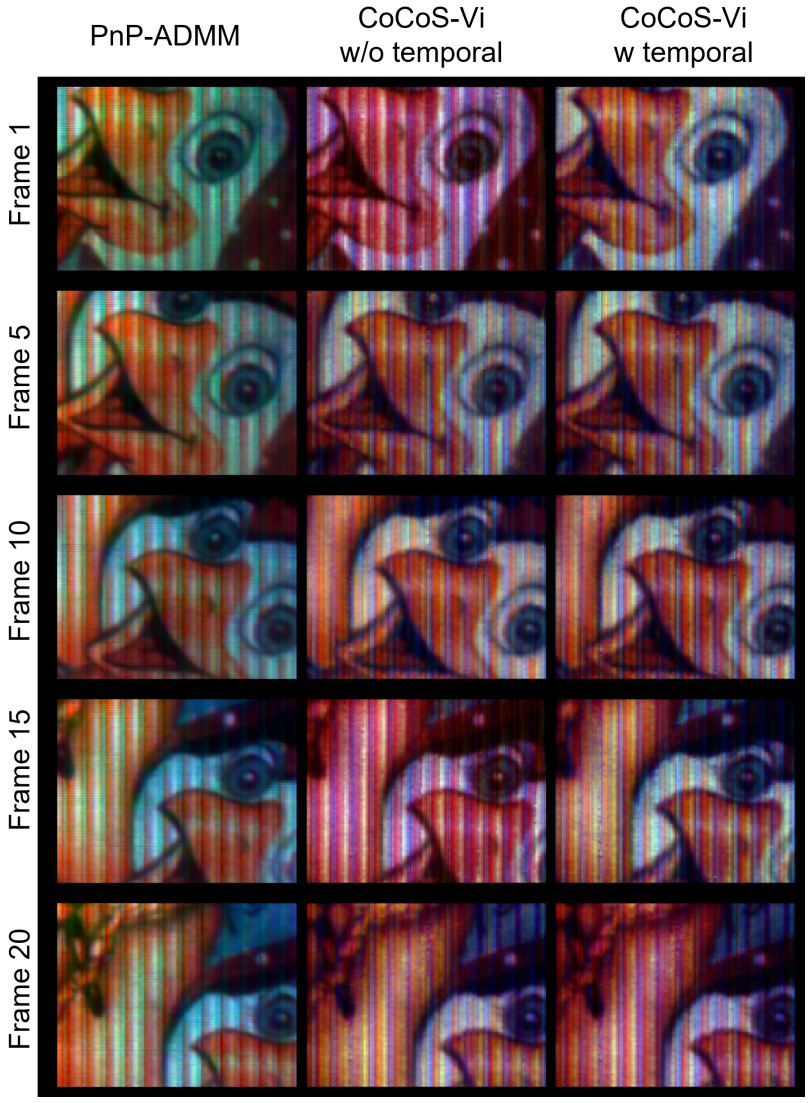


*Figure 37.* Five selected reconstructed frames in an RGB composite for (left column) PnP-ADMM, (middle column) CoCoS-Vi w/o temporal, and (right column) CoCoS-Vi w temporal.

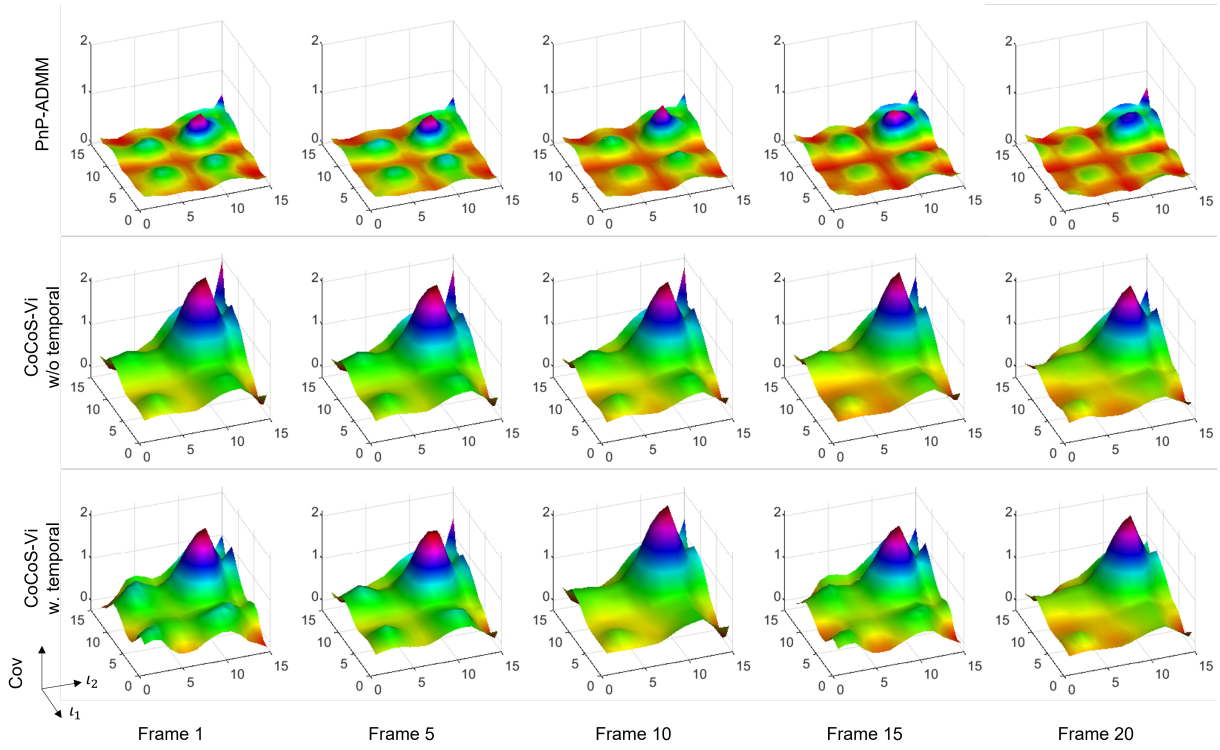


*Figure 38.* Five selected recovered spectral covariance matrices for the frames {1,5,10,15,20}, see Fig. 37. Here, for the PnP-ADMM approach, the covariance matrix is calculated directly from the reconstructed datacube.

.

These measurements were processed by the proximal gradient algorithm 4.1 with the hyperparameters $\tau = 0.001$ and 90 iterations and the video frames are reconstructed using (92) with 4 eigenvectors associated with the 4 largest eigenvalues. All simulations were conducted using an Intel Core i7 3960X 3.30 GHz processor with 32 GB RAM memory and Matlab 2021.b . We compare against the results obtained by using the CoCoS-Vi measurements with the uncorrelated covariance temporal algorithm proposed in (Monsalve et al., 2022b) (termed as CoCoS-Vi w/o termporal) and a PnP ADMM based algorithm that reconstruct the video without computing the covariance matrix (Yuan et al., 2020). Figure 36 depicts four out of the 15 representative re-

constructed bands and a RGB composition. Notice that the spatial reconstruction achieve by the "CoCoS-Vi w temporal" method is smoother than the obtained via "CoCoS-Vi w/o temporal" and "PnP-ADDM", respectively. Finally, Fig. 38 illustrates five reconstructed spectral CM for the frames shown in Fig. 36, Note that the reconstructed covariance matrices using PnP follows the same over-smoothed behaviour shown in simulations section. The execution time for the whole video reconstruction with 90 iterations in the PnP-ADMM, CoCoS-Vi w/o temporal, and CoCoS-Vi w temporal approach were measured as 556.42, 1.4, and 2.4 seconds, respectively. Note that, the CM-based approaches obtain a speedup of 397x (w/o temporal) and 231x (w temporal). This implies that the proposed method achieves 8.3 reconstructed frames per second.

### 4.4. Conclusions

This chapter introduced a compressive optical system (CoCoS-Vi) along with a sensing protocol to estimate the scene spectro-temporal covariance matrix from a single compressed measurement per frame. Further, the performance of the proposed sensing protocol is confirmed through a proof-of-concept implementation, which confirmed that the proposal represents an efficient alternative to estimate directly the CM. We validated that using a temporal correlation in the covariance improves the reconstruction of the spectral video without compromising the computational complexity. Additionally, the CM-based reconstruction methodology achieved an speed up of 231x against a traditional PnP-ADMMM algorithm.

# 5. Application of Covariance Matrix Recovery to Land Cover Estimation using Deep Learning. A Case Study at Valle de San José

*This chapter addresses the fourth and fifth objective of the thesis:*

- *To adapt a state-of-the-art algorithm to estimate the vegetation cover using sample statistics and random low-dimensional projections of hyperspectral images based on the proposed approach*
- *To verify the performance of the adapted algorithm comparing the accuracy in vegetation cover estimation with state-of-the-art algorithms*

## 5.1. Introduction

This chapter validates the proposed CCS recovery approach with a land cover classification scheme involving four components: (i) using the compressed SI modeling described in Chapter 2, (ii) the SI fast low-rank approximation based on the CCS technique also described in Chapter 2 (iii) the feature extractor learning, training a deep-learning model with several SI publicly available datasets, and (iv) the classification step, taking advantage of the support vector machine to classify with a few in-situ acquired samples. The proposed classification scheme is evaluated as a case study at Valle de San José in Santander, Colombia, considering the crop variety of the region and ease of access to the rural areas. Specifically, a set of SIs acquired with the Sentinel 2B satellite were employed for the compressive acquisition simulation. Further, the computational simulations were performed using a fast reconstruction from the estimated CV. The experiments

section analyzes the obtained accuracy when varying the compression ratio, showing that the CCS-based classification performs comparably to the classification using full data, while using a lower computational load.

## 5.2. Spectral Images Acquisition Details

This section describes the spectral image (SI) acquisition and pre-processing in the studio area, including details of the visit to the rural region and the analyzed classes.

**5.2.1. Area of studio.** The studio was conducted in Valle de San Jose, Santander, Colombia, presenting a mountainous geography characterized by the cultivation of coffee, sugar cane, and cattle pastures, coffee being the most significant crop with the highest presence. Figure 39 shows an RGB image of the selected region, centered on the WGS84 geographic coordinates of $6^o$ 590 1300 N, $73^o$ 400 6200 W.

The referenced image was acquired on September 10, 2022, using the Sentinel-2 VNIR sensor (Dechoz et al., 2015). For the study, we selected a subregion of $440 \times 680$ spatial pixels and 10 spectral channels covering the 490 nm to 2190 nm spectral range. Originally, the spatial resolution is of 10m for the 2,3,4 and 8 spectral channels; and 20m for the 5,6,7,11, and 12 spectral channels. Nonetheless, the spatial resolution was adjusted at 10m for all channels by sub-sampling the latter channels.

**5.2.2. Definition of the Classes.** The identification of the predominant classes was carried out by means of a visit in-situ in the area of the studio, analyzing the ground truth of the vegetation at some specific locations to create a land-cover inventory. In particular, the region contains a large agricultural area, mostly corresponding to coffee crops, followed by sugar cane,

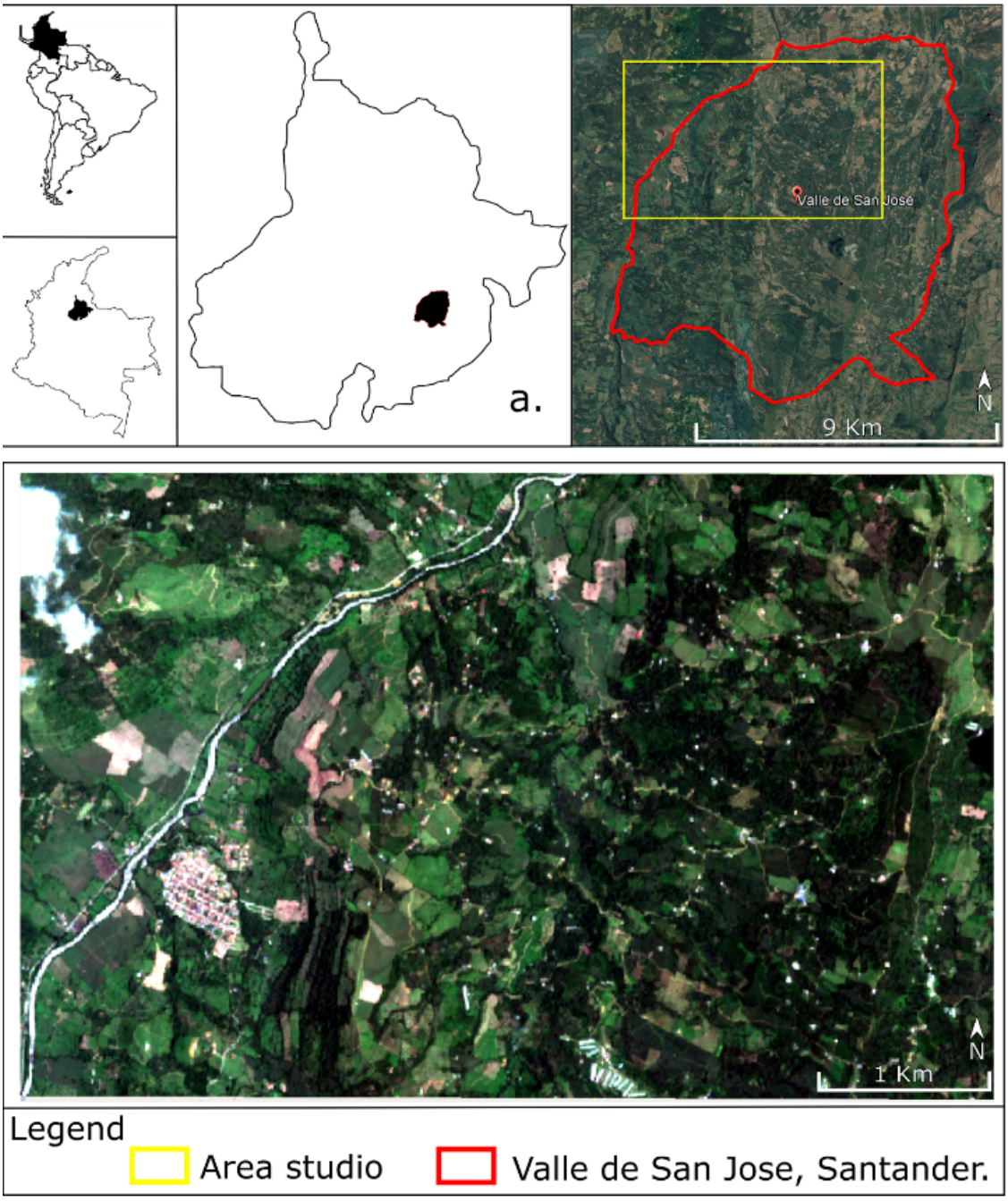


*Figure 39.* Visualization of the studio area located at $6^o$ 590 1300 N, $73^o$ 400 6200 W in Valle de San José, Santander, Colombia. The red line boxes the entire municipality. The yellow box limits the observed region, covering a mountainous area with agricultural vegetation.

cocoa, and pastures. Hence, we define the following five classes:

1. Coffee crops and trees class.
2. Agricultural vegetation class, excluding coffee.
3. Bare soil class, land preparation for future crops.
4. Urban areas class.
5. Water class.

Figure 40 shows the visit to the region and the geo-reference process of the selected classes and areas, by taking control points with a sub-meter GPS and photographs to identify the pixels in the Sentinel-2b SI. Through this visit, around 500 pixels per class were identified in the satellite image using the captured GPS points on-site.

Figure 41 shows some locations of the pixels whose spectral signatures were extracted for the training and testing process for each defined land-cover class. There, notice the high similarities between the spectral response of the defined classes, indicating that the classification of such image is a challenging task.

### 5.3. Proposed Classification Method

This section describes the mathematical details of the proposed scheme to perform the semi-supervised land cover classification based on compressive covariance sensing ( CCS) involving four components: (i) The modeling of the SI compressed version. (ii) The low-rank approximation of the SI based on CCS. (iii) The learning of a feature extractor with a convolutional neural network

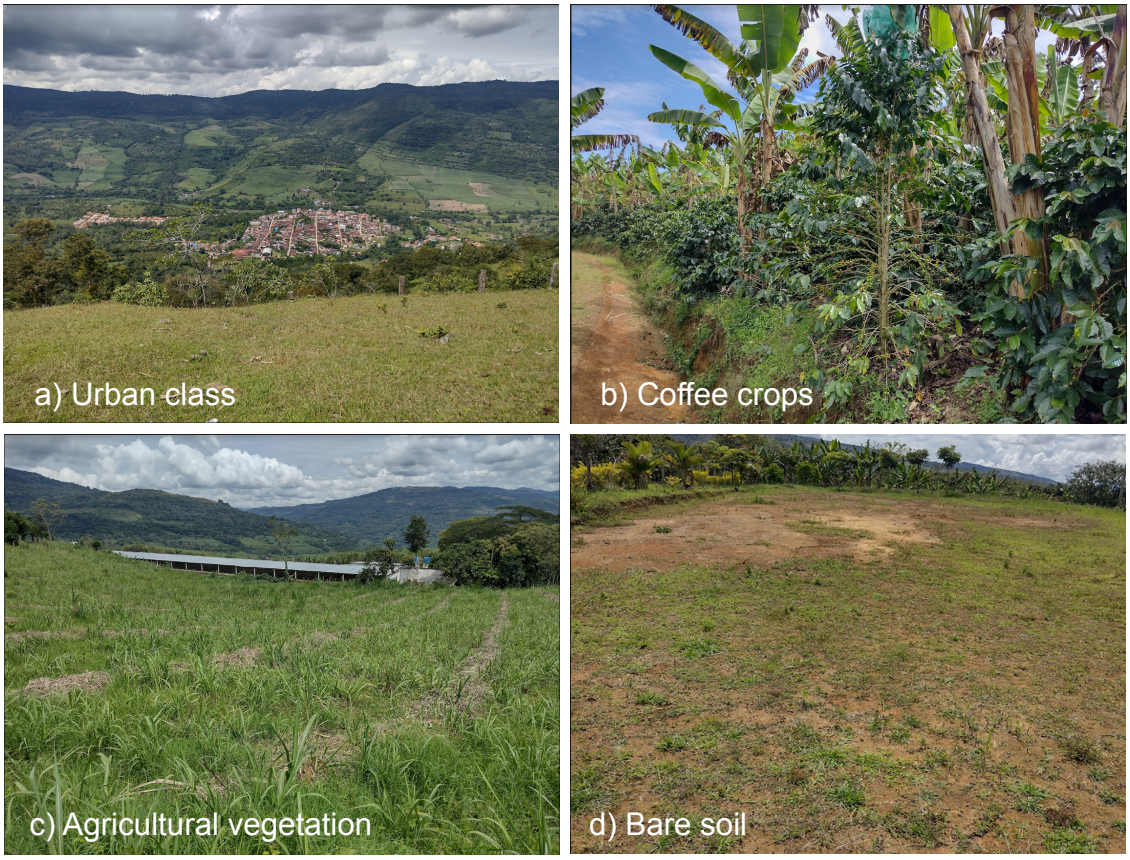


*Figure 40.* Visit in Valle de San José, taking the control checkpoints of the selected classes and areas.

(CNN) model. And (iv) a support vector machine (SVM) classifier. Each component is detailed in the following subsections.

**5.3.1. Compressive Spectral Imaging.** Let $\mathbf{F} \in \mathbb{R}^{l \times n}$, denote the matrix form of a SI with $l$ spectral bands and $n$ spatial pixels. The acquisition of a compressed version $\mathbf{Y} \in \mathbb{R}^{m \times n}$, with $m \ll n$ can be modelled by the following CSI forward model

$$\mathbf{Y} = \mathbf{P}^T \mathbf{F} + \mathbf{N}, \tag{93}$$

where $\mathbf{P} \in \mathbb{R}^{l \times m}$ denotes a random projection matrix, and $\mathbf{N} \in \mathbb{R}^{m \times n}$ models acquisition noise(Monsalve et al., 2022b,a).

**5.3.2. Compressive Covariance Sensing Recovery.** Given the compressed SI, CCS recovers the complete SI based on the CM, reducing the computational load of traditional CSI reconstruction algorithms. Nonetheless, in most cases the CM $\mathbf{S}$ is unknown and must be estimated from $\mathbf{Y}$. Therefore, this work is based on the fast CM estimation approach presented in (Monsalve et al., 2022b, 2021), which demonstrated that splitting the signal $\mathbf{F}$ into $p$ disjoint subsets $\mathbf{F}_i \in \mathbb{R}^{l \times n/p}$ and projecting them onto different subspaces $\mathbf{P}_i \in \mathbb{R}^{l \times m}$ allows to accurately estimate the CM $\mathbf{S}$ from compressed measurements by solving the optimization problem in (58)

After estimating the CM $\mathbf{S}$, a low-rank approximation of the SI is computed via the following preconditioned pseudo inverse problem

$$\tilde{\mathbf{F}} = (\mathbf{P}^T \mathbf{W})^{\dagger} \mathbf{Y}, \tag{94}$$

where $\tilde{\mathbf{F}} \in \mathbb{R}^{k \times n}$ denotes the SI low-rank approximation with the first $k$ PCA coefficients, and $\mathbf{W}$ is a matrix containing the first $k$ eigenvectors of $\mathbf{S}$(Fowler, 2009).

**5.3.3. Feature Extractor Learning using a Convolutional Neural Network .** The architecture of a CNN usually compresses the input into a latent space before applying a dense layer with a softmax to assign a class to the input. Literature on self-supervised learning has shown that such learned latent space can act as a descriptor or feature extractor where the model is fine-tuned for a specific task, useful in classification tasks with few available samples, as is our

case with few images captured in the area of the studio(Zhai et al., 2019; Hendrycks et al., 2019).

Unlike traditional pixel-wise classification approaches operating over single pixels, the proposed scheme learns the latent space using a patch-based classification approach, which extracts and labels spatial patches of the image, promoting spatial priors and smooth classifications (Makantasis et al., 2015). The used CNN architecture consists of two convolutional layers and two dense layers, a simple architecture for fast and accurate SI classification with few trainable parameters. Notice that the input of the CNN corresponds to the first PCA coefficients of each spatial patch, given the low-rank approximation estimated in the previous component, resulting in a reduction of the computational complexity of the process.

**5.3.4. Support Vector Machine Classifier.** The fine-tuning removes the last layer of the CNN model used to learn the feature extractor and trains the model with new classes and data, taking advantage of the SVM generalization capabilities when dealing with a few samples (Tang, 2013). Additionally, the computational complexity is also reduced since only the SVM needs to be trained using the latent projection of the signal generated by the CNN model.

Formally, let $g_\theta : \mathbb{R}^l \to \mathbb{R}^s$ be a CNN model whose last layer has been removed. Additionally, let $(\mathbf{f}_i, y_i)$ be a labeled pair of data for training. The corresponding optimization problem for the SVM is given by

$$\min_{\mathbf{u},b} \frac{1}{2}\|\mathbf{u}\|^2 + C\sum_{i=1}^{N} \max(1 - y_i(\mathbf{u}^T g_\theta(\mathbf{f}_i) + b), 0)^2, \tag{95}$$

where $\mathbf{u}$ contains the normal vectors to the hyperplane separating the classes (Tang, 2013).

Figure 42 illustrates and summarizes the proposed scheme to perform the semi-supervised land cover classification using CCS. In a), the input corresponds to compressed SI $\mathbf{Y}$ in (93). In b), the CM is estimated by solving the problem in (58), and a low-rank approximation of the SI is obtained using the matrix $\mathbf{W}$ containing the first $k$ eigenvectors of the estimated CM $\mathbf{S}^*$, following the CCS solution in (94). Subsequently, in c), a patch-based classification CNN is used to learn a latent space used as the feature extractor, where the CNN architecture consists of two convolutional layers and two dense layers, trained with literature SI datasets covering various classes. Finally, in d), an SVM replaces the last layer of the CNN to fine-tune the classification task over the few samples acquired at Valle de San José using the training process according to the solution of (95).

## 5.4. RESULTS AND DISCUSSION

This section describes the experiments carried out to evaluate the performance of the proposed land-cover classification scheme. Specifically, the classification scheme is used over simulated data to validate the effectiveness; and then is used to classify the acquired satellite image with the defined classes in Valle de San José.

**5.4.1. Experimental Setup.** The experiments use four SI datasets. The Indian Pines hyperspectral dataset, acquired by the AVIRIS sensor from the Northwestern Indian Pines test site in June 1992 with $145 \times 145$ spatial pixels and 200 spectral bands (Baumgardner et al., 2015). The Pavia University and Pavia Center datasets, acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over Pavia, Northern Italy with $610 \times 340$ spatial pixels and 103 spectral bands (, GIC). And Salinas dataset, a high spatial resolution image collected by the 224-band AVIRIS sensor over Salinas Valley with $512 \times 217$ spatial

pixels (, GIC).

The performance of the proposed methodology is evaluated using four metrics. First, the Average Accuracy (AA) computes the average accuracy for each class useful to evaluate unbalanced datasets. The overall accuracy (OA) measures the correct number of pixels across the classes. The F1 score is used for the binary classification problems as a ratio between precision and recall, hence the F1 score is computed as the average of the F1 for each class. Finally, kappa statistics is also used since it considers that assigning labels randomly has a certain degree of accuracy, making this metric a robust way to measure classification performance (Hinojosa et al., 2018b).

**5.4.2. Feature Extractor Training.** The training process to learn the feature extractor uses three of the four datasets, addressing a completely supervised training (University of Pavia, Salinas, Indian Pines). The model is trained progressively using each dataset, applying a PCA, and adjusting the model with the following corresponding classes.

For each dataset do:

- Apply PCA to obtain a dimensionality-reduced version using six components.
- Adjust the dense layer to match the number of classes in the dataset.
- Run 1000 iterations to retrain the feature extractor with the current classes.

The process is repeated until the model converges, no matter which image is used. The intuition of this process is that the model can learn to extract characteristics of the spectral signatures independently of the classes.

**5.4.3. Classification with Pavia Center dataset.** The remaining dataset (Pavia Center) is used for semi-supervised training using the SVM as the classification layer keeping the CNN model weights fixed, used as the feature extractor. For the training, 100 pixels of each class were used to fine-tune the CNN model and to train the SVM classifier in the experiments. This experiment tests two scenarios, using full data with no compression other than the PCA dimensionality reduction and applying a random compression on the data using the CCS approach as described in Section 5.3 acquiring only 10% of the data.

Table 5 shows the results for different scenarios labeled as follows: using full data and the CNN model with a dense layer and a Sigmoid activation as a classifier (CNN-full); using full data and the CNN model with the SVM as a classifier (SVM-full); using compressed data and the CNN model with a dense layer and a Sigmoid activation as a classifier (CNN-CCS); and using compressed data and the CNN model with the SVM as a classifier (SVM-CCS).

There, it can be seen that the classification performance when using compressed versions of the dataset achieves comparable results in terms of classification accuracy with those obtained using full data. Furthermore, using the SVM achieves comparable or even improves the Kappa metric and F1 score, indicating that the SVM approach provides a better classification agreement and balance between the recall and precision metrics.

Figure 43 shows a visual comparison between the four evaluated scenarios. It can be seen that the results of the compressed datasets are comparable to those obtained using full data, especially when the SVM classifier is used as visualized in the boxed region where the CNN scenario

presents artifacts.

**5.4.4. Experiments with Valle de San José data.** The Valle de San José data was compressed following equation (93) using 60% of the information. The classification results are compared against the results obtained using full data. Figure 44 illustrates the obtained classifications over the real data, where using the compressed measurements achieves comparable results with those obtained with full data. Since no ground truth is available, we present a qualitative analysis of the predicted classes. Based on the knowledge of the zone and the visit in situ, we observed a high correlation with the expected distribution of the classes. The (A) region in figure 44 corresponds to the location of the town which is correctly predicted. Region B corresponds to bare soil, it is interesting that using the compressed measurements, the algorithm was able to predict it correctly meanwhile, using full data misclassified it. Finally, region C was predicted as water in both compressed measurements, but there is no water in that zone. Overall, using an SVM with compressed data exhibits a better performance than using only a CNN model.

| Model | AA(%) | OA(%) | Kappa | F1 score |
|---|---|---|---|---|
| CNN-Full | $98.6 \pm 0.2$ | $93.8 \pm 0.8$ | $91.5 \pm 1$ | $87.0 \pm 1.3$ |
| SVM- Full | $98.6 \pm 0.2$ | $93.8 \pm 0.9$ | $91.5 \pm 1.3$ | $86.7 \pm 1.8$ |
| CNN-CCS | $98.5 \pm 0.4$ | $93.3 \pm 1.8$ | $90.6 \pm 2.5$ | $85.2 \pm 2.4$ |
| SVM-CCS | $98.6 \pm 0.1$ | $93.8 \pm 0.8$ | $91.3 \pm 1.1$ | $85.8 \pm 1.1$ |

Table 5
*Classification quantitative results Pavia Center Image.*

### 5.5. Conclusions

This work presented a land cover classification scheme based on the compressive covariance sensing technique. In particular, the classification scheme was developed to be used in a case study over a region in Vallé de San José, Santander, Colombia. For this, we carried out a visit in-situ to the region, analyzing the vegetation present at different locations and defining the predominant classes to build a land cover inventory containing the corresponding spectral signatures. Furthermore, we used a spectral image of the region acquired by the Sentinel-2 VNIR sensor to be used in the classification. The developed classification scheme involves a deep-learning feature extractor and a support vector machine classifier, considering the few observed samples of the different classes of interest in the real study case. We conducted experiments over different configurations, showing the effectiveness of the proposed classification scheme to discriminate the predominant classes in the area of the studio and, more importantly, achieving a comparable performance of literature methods using complete data at a reduced computational load.

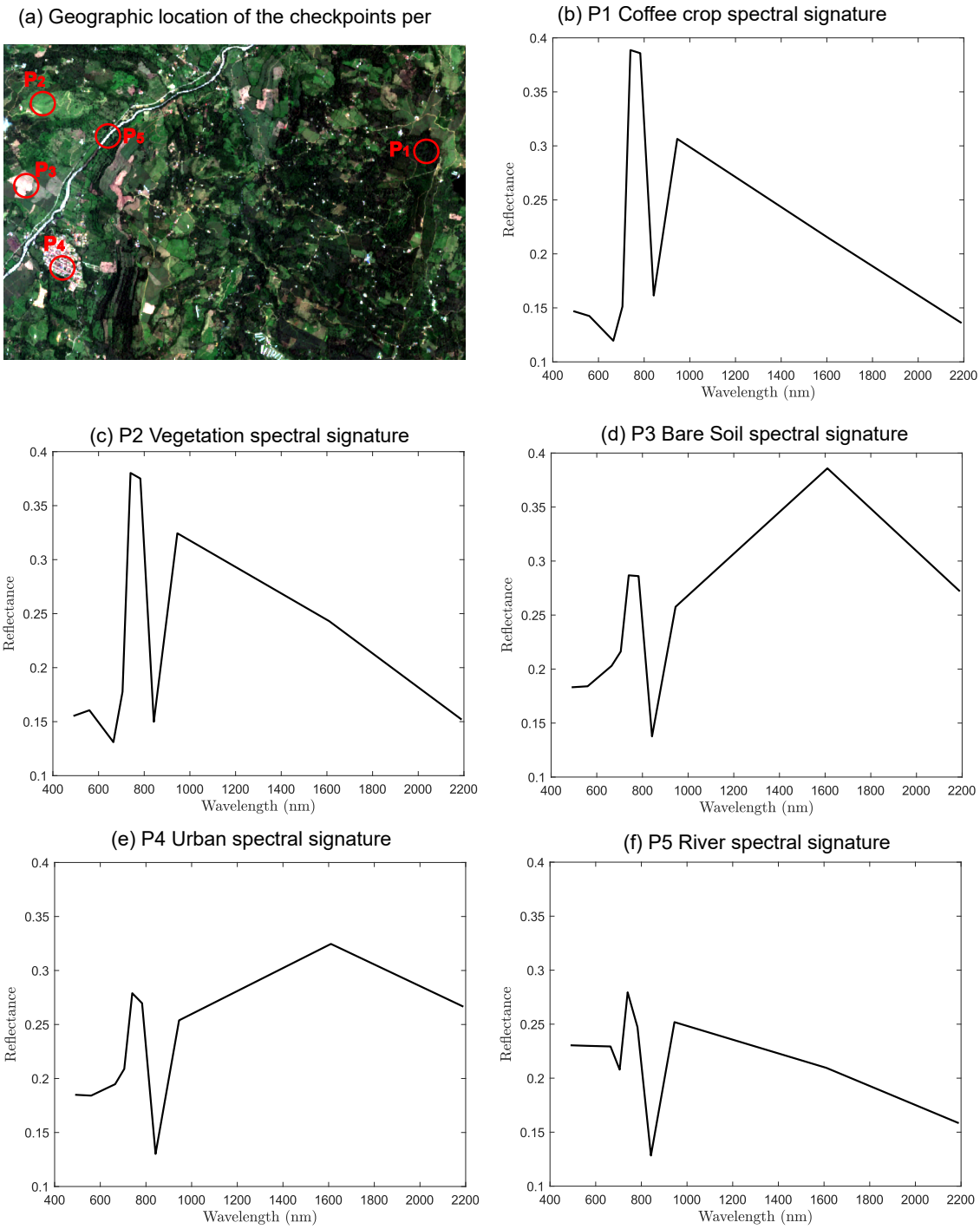


*Figure 41.* Average spectral response of the five land-cover classes defined in the present study. (*a*) Location of the main areas where the pixels can be found on the Sentinel-2 image (10 m). (*b*) Coffee crop, $P_1$. (*c*) Agricultural vegetation without coffee, $P_2$. (*d*) Bare soil, $P_3$. (*e*) Urban areas, $P_4$., and (*f*) River, $P_5$.

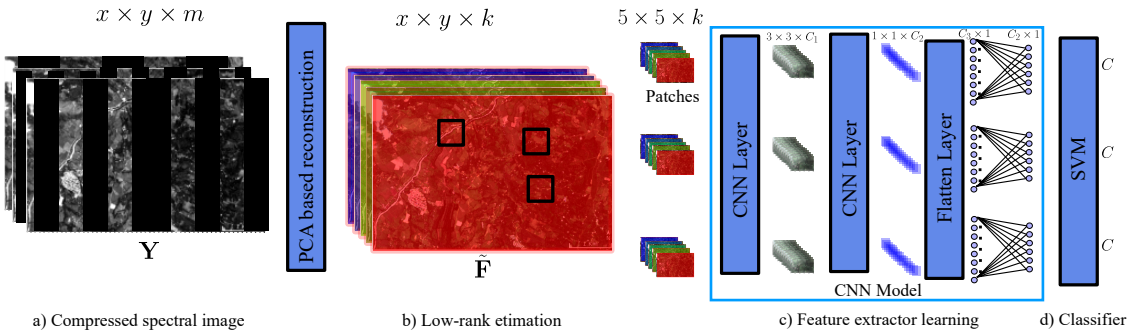


*Figure 42.* Schematic representation of the semisupervised patch-based classification approach from compressed measurements. a) The input consists of a compressed spectral image. b) The low-rank approximation with just k principal components, estimated following the compressive covariance sensing technique. c) The convolutional neural network (CNN) model architecture is used to learn the feature extractor. d) The support vector machine classifier replaces the CNN model's last layer to classify the few samples at Valle de San José.

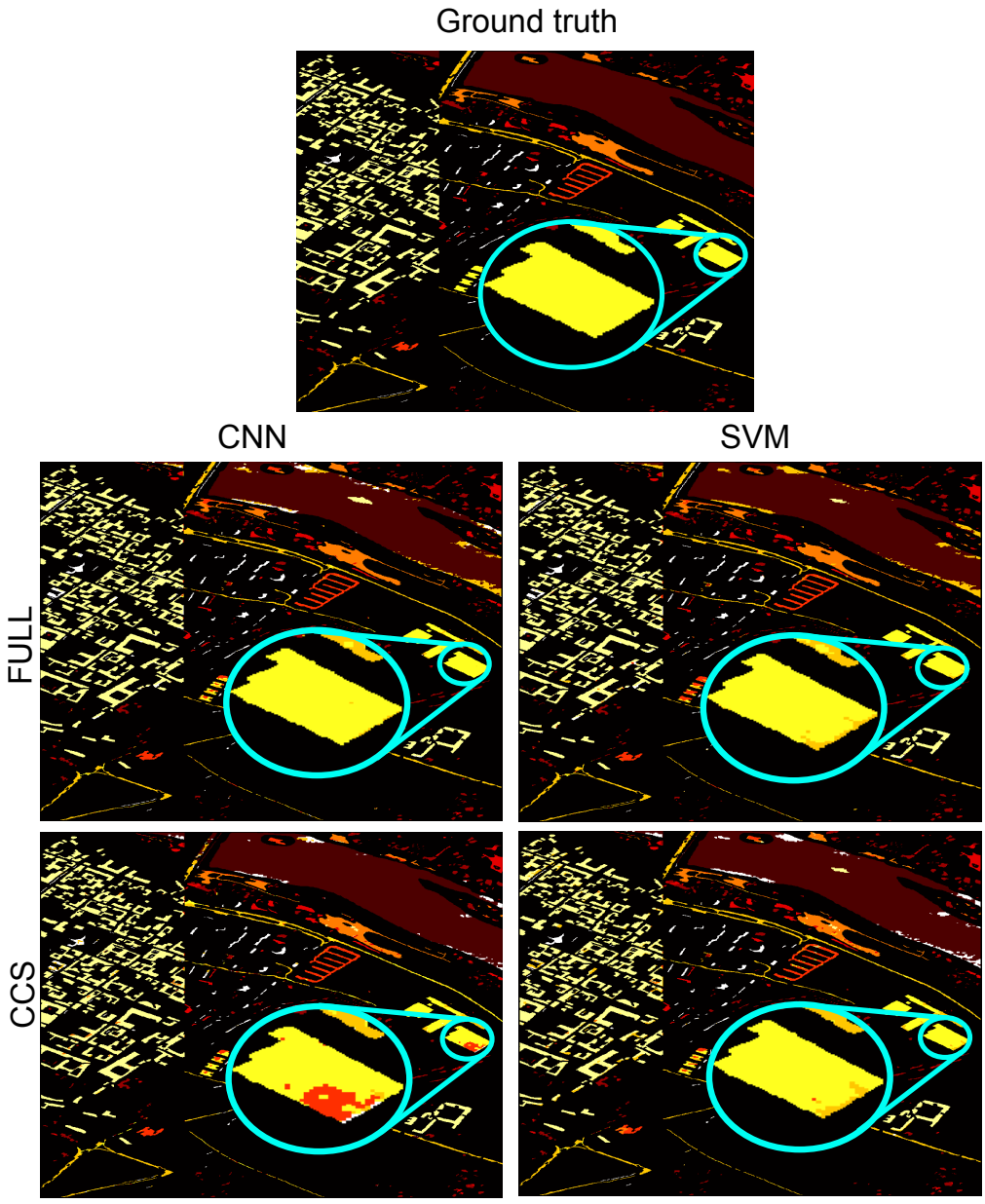


*Figure 43.* Visual comparison of the classification results. Note that the black regions are unlabeled zones hence, it was masked out in the results for interpretability

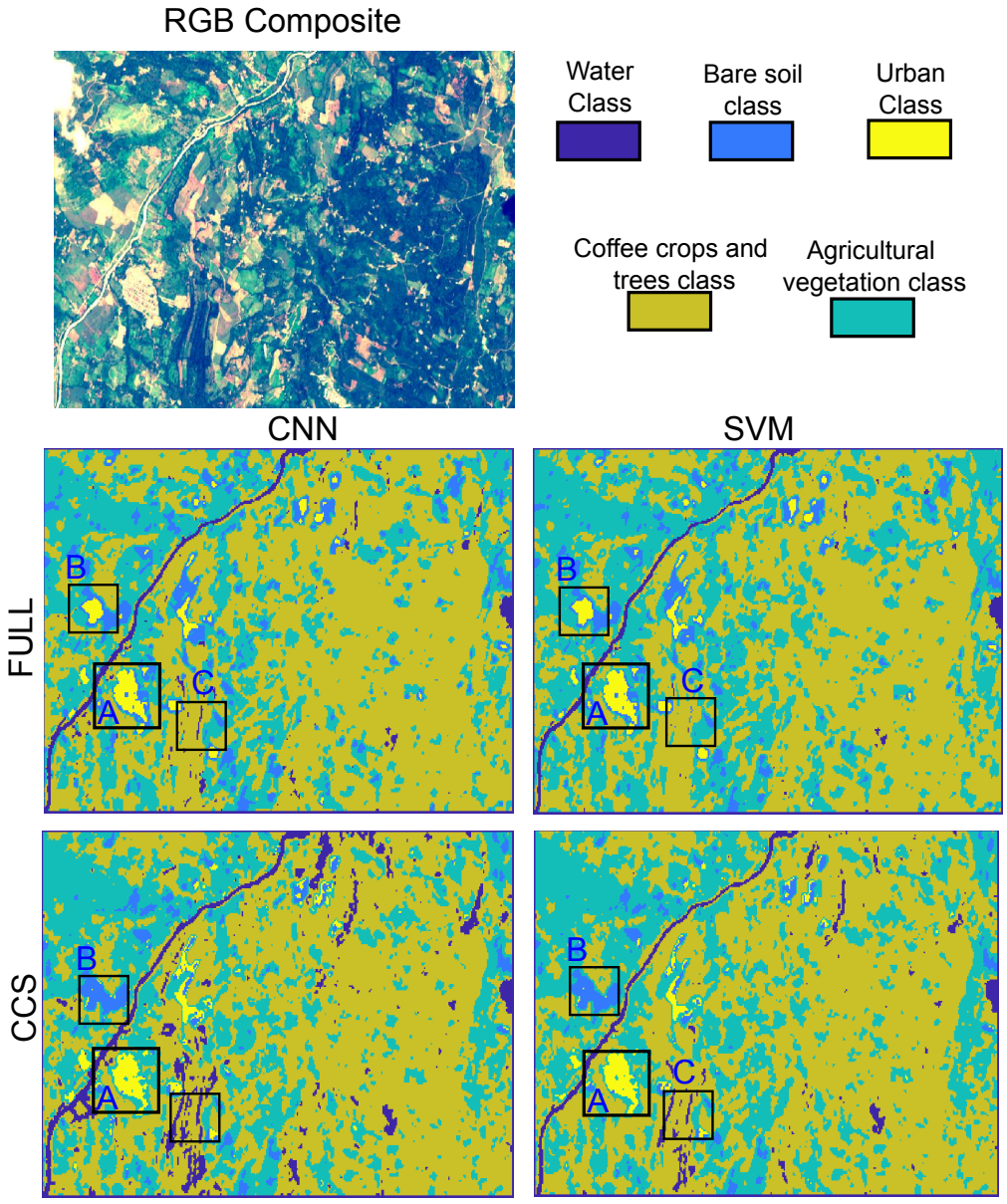


*Figure 44.* Visual comparison of the results obtained using the data from valle de san josé

# 6. Conclusions

This dissertation proposed a framework to improve compressive spectral imaging applications using the first and second statistical moments. This framework includes the estimation of the mean and covariance matrix (CM), its use in the acquisition process, an extension to be applied in a spectral video setup along with an optical architecture for this purpose, and finally, an application to land cover estimation using deep learning and the recovered covariance.

The hypothesis of this dissertation was: *The first and second sample statistical moments of a dataset can be accurately estimated directly from low-dimensional random projections and they can be used to improve the reconstruction of the high dimensional data using compressive spectral imaging and principal component analysis based techniques*. Chapters 2 to 5 introduced 4 algorithms and its respective experimental validations that provide sufficient evidence to support that the hypothesis is true.

Chapter 2 introduced an algorithm to design the sensing matrix in compressive spectral imaging using the covariance matrix. The algorithm uses a greedy strategy to maximize the variance of the projected signal in the resulting binary subspace. This chapter proved that by maximizing the variance retained by a few binary vectors, the reconstruction performance improves up to 3 dB in terms of PSNR compared to random matrices. Additionally, the proposed approach preserves the variance of the compressed data in a better way than other methods, such as Q-PCA. However, the CM must be known either as priori information or by estimating it in a addaptive manner.

Therefore, Chapter 3 proposed a fast algorithm to recover a low-rank approximation of the CM. This algorithm used the idea that the interception of multiple approximations in different subspaces improves the reconstruction. The algorithm is based on a projected gradient method and used an Armijo search strategy to speed up the convergence. This chapter also presented theoretical convergence guarantees and error analysis. The algorithm was tested using binary, normally distributed, and uniformly distributed matrices achieving the best performance with normally distributed matrices. The algorithm performance was tested against two state-of-the-art algorithms achieving better results in most scenarios and faster convergence in terms of computing time. For the CPPCA algorithm, the proposed algorithm achieved up to 2 orders of magnitude of improvement in terms of MSE but is one order of magnitude slower. In the case of SpeCA the proposed method was up to one order of magnitude faster but achieved comparable results in terms of MSE. This Chapter also presented a novel optical architecture that can capture the structured measurements that were designed theoretically, validating the theoretical findings.

Motivated by the previous chapter's findings, Chapter 4 proposed an extension of the algorithm to take advantage of the temporal correlation allowing to use of it in a spectral video setup. The temporal correlation was exploited by adding a low-rank restriction in the temporal dimension. However, the multishot approach used in the last chapter is infeasible in the video; hence, Chapter 4 also proposed a new optical architecture that captures multiple random projections using a lens-let array. The validation was performed in the optical laboratory achieving up to two orders of magnitude of improvement in terms of NMSE and a speedup of 231x in comparison to an ADMM algorithm. This speedup enabled to use of the proposed algorithm in a real-time scenario achieving

up to 8.3 reconstructed frames per second.

Chapters 3 and 4 proved that the proposed algorithms behaves well in terms of MSE and PSNR. However, to fully validate the algorithm it must be tested in a higher level task such as classification. To this end, Chapter 5 used the algorithm from Chapter 3 to reconstrcut spectral images from satellital platforms and perform a land cover estimation task. This was achieved by using an state-of-the-art classification algorithm based on deep learning using the reconstructed data. The results showed that the classification using compressed measurements achieved comparable results to those obtained using full data and in some scenarios achieving the same performance. In general, the proposed framework supports the hypothesis of the dissertation and demonstrates its potential for improving compressive spectral imaging applications.

## Bibliography

(2022). Eliminating temporal illumination variations in whisk-broom hyperspectral imaging. *International Journal of Computer Vision*, 130:1310–1324.

Afonso, M. V., Bioucas-Dias, J. M., and Figueiredo, M. A. T. (2010). Fast Image Recovery Using Variable Splitting and Constrained Optimization. *IEEE Transactions on Image Processing*, 19(9):2345–2356.

Al-Sarayreh, M., Reis, M. M., Yan, W. Q., and Klette, R. (2020). Potential of deep learning and snapshot hyperspectral imaging for classification of species in meat. *Food Control*, 117:107332.

Anaraki, F. P. and Hughes, S. M. (2014). Efficient recovery of principal components from compressive measurements with application to Gaussian mixture model estimation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2332–2336. IEEE.

Arce, G. R., Brady, D. J., Carin, L., Arguello, H., and Kittle, D. S. (2013). Compressive coded aperture spectral imaging: An introduction. *IEEE Signal Processing Magazine*, 31(1):105–115.

Arce, G. R., Brady, D. J., Carin, L., Arguello, H., and Kittle, D. S. (2014a). Compressive Coded Aperture Spectral Imaging: An Introduction. *IEEE Signal Processing Magazine*, 31(1):105–115.

Arce, G. R., Brady, D. J., Carin, L., Arguello, H., and Kittle, D. S. (2014b). Compressive Coded

Aperture Spectral Imaging: An Introduction. *IEEE Signal Processing Magazine*, 31(1):105–115.

Arguello, H. and Arce, G. R. (2012a). Restricted Isometry Property in coded aperture compressive spectral imaging. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 716–719. IEEE, IEEE.

Arguello, H. and Arce, G. R. (2012b). Restricted isometry property in coded aperture compressive spectral imaging. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 716–719.

Arguello, H. and Arce, G. R. (2014). Colored coded aperture design by concentration of measure in compressive spectral imaging. *IEEE Transactions on Image Processing*, 23(4):1896–1908.

Arguello, H., Correa, C. V., and Arce, G. R. (2013). Fast lapped block reconstructions in compressive spectral imaging. *Appl. Opt.*, 52(10):D32–D45.

Arguello, H., Pinilla, S., Peng, Y., Ikoma, H., Bacca, J., and Wetzstein, G. (2021). Shift-variant color-coded diffractive spectral imaging system. *Optica*, 8(11):1424–1434.

Balakrishnama, S. and Ganapathiraju, A. (1998). Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18(1998):1–8.

Baumgardner, M. F., Biehl, L. L., and Landgrebe, D. A. (2015). 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3.

Beck, A. (2017). Chapter 10: The Proximal Gradient Method. *First-Order Methods in Optimization*, pages 269–329.

Besson, O., Bidon, S., and Tourneret, J. Y. (2008a). Bounds for estimation of covariance matrices from heterogeneous samples. *IEEE Transactions on Signal Processing*, 56(7 II):3357–3362.

Besson, O., Bidon, S., and Tourneret, J. Y. (2008b). Covariance matrix estimation with heterogeneous samples. *IEEE Transactions on Signal Processing*, 56:909–920.

Bioucas-Dias, J. M., Cohen, D., and Eldar, Y. C. (2014). Covalsa: Covariance estimation from compressive measurements using alternating minimization. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 999–1003.

Bioucas-Dias, J. M., Cohen, D., and Eldar, Y. C. (2014a). Covalsa: Covariance estimation from compressive measurements using alternating minimization. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 999–1003.

Bioucas-Dias, J. M., Cohen, D., and Eldar, Y. C. (2014b). Covalsa: Covariance estimation from compressive measurements using alternating minimization. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 999–1003.

Bioucas-Dias, J. M. and Nascimento, J. M. (2008). Hyperspectral subspace identification. *IEEE Transactions on Geoscience and Remote Sensing*, 46(8):2435–2445.

Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N. M., and Chanussot,

J. (2013). Hyperspectral remote sensing data analysis and future challenges. *Geoscience and Remote Sensing Magazine, IEEE*, 1(2):6–36.

Blanco, G., Perez, J., Monsalve, J., Marquez, M., Esnaola, I., and Arguello, H. (2021). Single snapshot system for compressive covariance matrix estimation for hyperspectral imaging via lenslet array. In *2021 XXIII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, pages 1–5.

Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arxiv*.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982.

Candès, E. and Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985.

Candes, E. and Wakin, M. (2008). An Introduction To Compressive Sampling. *IEEE Signal Processing Magazine*, 25(2):21–30.

Candes, E. J. and Tao, T. (2005). Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215.

Candiago, S., Remondino, F., De Giglio, M., Dubbini, M., and Gattelli, M. (2015). Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images. *Remote Sensing*, 7(4):4026–4047.

Cao, X., Yue, T., Lin, X., Lin, S., Yuan, X., Dai, Q., Carin, L., and Brady, D. J. (2016). Computational Snapshot Multispectral Cameras: Toward dynamic capture of the spectral world. *IEEE Signal Processing Magazine*, 33(5):95–108.

Chen, C., Li, W., Tramel, E. W., and Fowler, J. E. (2014). Reconstruction of hyperspectral imagery from random projections using multihypothesis prediction. *IEEE Transactions on Geoscience and Remote Sensing*, 52:365–374.

Chen, Y., Wiesel, A., Eldar, Y. C., and Hero, A. O. (2010). Shrinkage algorithms for mmse covariance estimation. *Trans. Sig. Proc.*, 58(10):5016–5029.

Cline, R. E. and Plemmons, R. J. (1976). $l_2 $-Solutions to Underdetermined Linear Systems. *SIAM Review*, 18(1):92–106.

Correa, C. V., Arguello, H., and Arce, G. R. (2016a). Spatiotemporal blue noise coded aperture design for multi-shot compressive spectral imaging. *J. Opt. Soc. Am. A*, 33(12):2312–2322.

Correa, C. V., Hinojosa, C. A., Arce, G. R., and Arguello, H. (2016b). Multiple snapshot colored compressive spectral imager. *Optical Engineering*, 56(4):041309.

Correa, C. V., Hinojosa, C. A., Arce, G. R., and Arguello Sr, H. (2016c). Multiple snapshot colored compressive spectral imager. *Optical Engineering*, 56(4):041309.

Correa, C. V., Hinojosa, C. A. A., Arce, G. R., and Arguello, H. (2016d). Multiple snapshot colored compressive spectral imager. *Optical Engineering*, 56(4):10.

Dasgupta, S. and Gupta, A. (2003). An Elementary Proof of a Theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65.

Dechoz, C., Poulain, V., Massera, S., Languille, F., Greslou, D., De Lussy, F., Gaudel, A., L'Helguen, C., Picard, C., and Trémas, T. (2015). Sentinel 2 global reference image. In *Image and Signal Processing for Remote Sensing XXI*, volume 9643, page 96430A. International Society for Optics and Photonics.

Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

Figueiredo, M., Nowak, R., and Wright, S. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics Signal Process*, 1(4):586–597.

Foley, C., Monakhova, K., Yanny, K., and Waller, L. (2022). Spectral defocuscam: hyperspectral imaging using defocus and a spectral filter array. In *Imaging and Applied Optics Congress 2022 (3D, AOA, COSI, ISA, pcAOP)*, page CF2C.1. Optica Publishing Group.

Foucart, S. and Rauhut, H. (2013). An invitation to compressive sensing. In *A mathematical introduction to compressive sensing*, pages 1–39. Springer.

Fowler, J. (2009). Compressive-projection principal component analysis. *IEEE Transactions on Image Processing*, 18:2230–2242.

Fowler, J. E. and Du, Q. (2011). Reconstructions from compressive random projections of hyperspectral imagery. In *Optical Remote Sensing*, pages 31–48. Springer.

Fowler, J. E. and Du, Q. (2012). Anomaly detection and reconstruction from random projections. *IEEE Transactions on Image Processing*, 21(1):184–195.

Galvis-Carreño, D. F., Mejía-Melgarejo, Y. H., and Arguello-Fuentes, H. (2014). Efficient reconstruction of Raman spectroscopy imaging based on compressive sensing. *DYNA*, 81(188):116–124.

Gao, Z., Shao, Y., Xuan, G., Wang, Y., Liu, Y., and Han, X. (2020). Real-time hyperspectral imaging for the in-field estimation of strawberry ripeness with deep learning. *Artificial Intelligence in Agriculture*, 4:31–38.

Gehm, M. E., John, R., Brady, D. J., Willett, R. M., and Schulz, T. J. (2007a). Single-shot compressive spectral imaging with a dual-disperser architecture. *Optics Express*, 15(21):14013–14027.

Gehm, M. E., John, R., Brady, D. J., Willett, R. M., and Schulz, T. J. (2007b). Single-shot compressive spectral imaging with a dual-disperser architecture. *Optics Express*, 15(21):14013–14027.

Gelvez, T. and Arguello, H. (2021). Nonlocal low-rank abundance prior for compressive spectral image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 59:415–425.

Gelvez, T., Rueda, H., and Arguello, H. (2017). Joint sparse and low rank recovery algorithm for compressive hyperspectral imaging. *Applied optics*, 56(24):6785–6795.

(GIC), C. I. G. Hyperspectral remote sensing scenes.

Girshick, R. (2015). Fast r-cnn. In *International Conference on Computer Vision (ICCV)*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *arxiv*.

Grigoriadis, K., Frazho, A., and Skelton, R. (1994). Application of alternating convex projection methods for computation of positive Toeplitz matrices. *IEEE Trans. on Signal Processing*, 42(7):1873–1875.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32.

Hinojosa, C., Bacca, J., and Arguello, H. (2018a). Coded aperture design for compressive spectral subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1589–1600.

Hinojosa, C., Bacca, J., and Arguello, H. (2018b). Coded aperture design for compressive spectral subspace clustering. *IEEE Journal on Selected Topics in Signal Processing*, 12:1589–1600.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arxiv*.

Hu, W.-C. and Hsu, J.-F. (2013). Automatic spectral video matting. *Pattern Recognition*, 46:1183–1194.

Iusem, A. N. (2003). On the convergence properties of the projected gradient method for convex optimization. *Computational & Applied Mathematics*, 22(1):37–52.

Jia, Z. and Stewart, G. W. (2000). An analysis of the rayleigh-ritz method for approximating eigenspaces. *MATHEMATICS OF COMPUTATION*, 70:637–647.

Jiang, B., Ma, S., and Zhang, S. (2015). Tensor principal component analysis via convex optimization. *Mathematical Programming*, 150(2):423–457.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Contemporary Mathematics*, volume 26, pages 189–206. American Mathematical Society.

Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R. (2008). Generalized power method for sparse principal component analysis. *Neural Computation*, 21(11):3179–3213.

Ke, J., Ashok, A., and Neifeld, M. A. (2010). Object reconstruction from adaptive compressive measurements in feature-specific imaging. *Applied optics*, 49(34):H27–H39.

Ke, J. and Lam, E. Y. (2016). Fast compressive measurements acquisition using optimized binary sensing matrices for low-light-level imaging. *Optics Express*, 24(9):9869.

Khader, A., Yang, J., and Xiao, L. (2022). Nmf-dunet: Nonnegative matrix factorization inspired

deep unrolling networks for hyperspectral and multispectral image fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:5704–5720.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90.

Kwak, N. (2008). Principal Component Analysis Based on L1-Norm Maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1672–1680.

Leon-Lopez, K. M., Mouret, F., Arguello, H., and Tourneret, J. Y. (2022). Anomaly detection and classification in multispectral time series based on hidden markov models. *IEEE Transactions on Geoscience and Remote Sensing*, 60.

Levenson, R. M. and Mansfield, J. R. (2006). Multispectral imaging in biology and medicine: slices of life. *Cytometry part A*, 69(8):748–758.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., and Wei, X. (2022). Yolov6: A single-stage object detection framework for industrial applications. *arxiv*.

Li, W. and Fowler, J. E. (2011). Decoder-side dimensionality determination for compressive-projection principal component analysis of hyperspectral data. In *2011 18th IEEE International Conference on Image Processing*, pages 321–324. IEEE.

Li, W., Prasad, S., and Fowler, J. E. (2013a). Classification and reconstruction from random

projections for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 51(2):833–843.

Li, W., Prasad, S., and Fowler, J. E. (2013b). Integration of spectral–spatial information for hyperspectral image reconstruction from compressive random projections. *IEEE Geoscience and Remote Sensing Letters*, 10(6):1379–1383.

Lin, B., Tao, X., and Lu, J. (2020). Hyperspectral image denoising via matrix factorization and deep prior regularization. *IEEE Transactions on Image Processing*, 29:565–578.

Lin, X., Liu, Y., Wu, J., and Dai, Q. (2014a). Spatial-spectral encoded compressive hyperspectral imaging. *ACM Trans. Graph.*, 33(6):233:1–233:11.

Lin, X., Liu, Y., Wu, J., and Dai, Q. (2014b). Spatial-spectral encoded compressive hyperspectral imaging. *ACM Trans. Graph.*, 33(6):233:1———233:11.

Liu, J. and Zhang, J. (2014). Spectral unmixing via compressive sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11):7099–7110.

Liu, T., Li, Y. F., Liu, H., Zhang, Z., and Liu, S. (2019). Risir: Rapid infrared spectral imaging restoration model for industrial material detection in intelligent video systems. *IEEE Transactions on Industrial Informatics*, pages 1–1.

Lu, G. and Fei, B. (2014). Medical hyperspectral imaging: a review. *Journal of biomedical optics*, 19(1):10901.

Makantasis, K., Karantzalos, K., Doulamis, A., and Doulamis, N. (2015). Deep supervised learning for hyperspectral data classification through convolutional neural networks. volume 2015-November, pages 4959–4962. IEEE.

Manolakis, D., Lockwood, R., Cooley, T., and Jacobson, J. (2009). Hyperspectral detection algorithms: use covariances or subspaces? In Shen, S. S. and Lewis, P. E., editors, *Imaging Spectrometry XIV*, volume 7457, page 74570Q. International Society for Optics and Photonics, SPIE.

Manolakis, D., Marden, D., and Shaw, G. a. (2003). Hyperspectral Image Processing for Automatic Target Detection Applications. *Lincoln Laboratory Journal*, 14(1):79–116.

Markas, T. and Reif, J. (1993). Multispectral image compression algorithms. In *[Proceedings] DCC '93: Data Compression Conference*, pages 391–400. IEEE Comput. Soc. Press.

Marquez, M., Lai, Y., Liu, X., Jiang, C., Zhang, S., Arguello, H., and Liang, J. (2022). Deep-learning supervised snapshot compressive imaging enabled by an end-to-end adaptive neural network. *IEEE Journal of Selected Topics in Signal Processing*, 16(4):688–699.

Marquez, M., Mejia, Y., and Arguello, H. (2017). Compressive spectral image super-resolution by using singular value decomposition. *Optics Communications*, 404:163–168.

Marquez, M., Meza, P., Arguello, H., and Vera, E. (2019). Compressive spectral imaging via deformable mirror and colored-mosaic detector. *Optics express*, 27(13):17795–17808.

Marquez, M., Meza, P., Rojas, F., Arguello, H., and Vera, E. (2021a). Snapshot compressive spectral depth imaging from coded aberrations. *Opt. Express*, 29(6):8142–8159.

Marquez, M., Meza, P., Rojas, F., Arguello, H., and Vera, E. (2021b). Snapshot compressive spectral depth imaging from coded aberrations. *Opt. Express*, 29(6):8142–8159.

Marquez, M., Rueda, H., Rojas, F., and Arguello, H. (2020a). Compressive spectral imaging via virtual side information. In *Imaging and Applied Optics Congress*, page JW2A.34. Optica Publishing Group.

Marquez, M., Rueda-Chacon, H., and Arguello, H. (2020b). Compressive spectral light field image reconstruction via online tensor representation. *IEEE Transactions on Image Processing*, 29:3558–3568.

Marquez, M., Rueda-Chacon, H., and Arguello, H. (2020c). Compressive spectral light field image reconstruction via online tensor representation. *IEEE Transactions on Image Processing*, 29:3558–3568.

Marquez, M., Rueda-Chacon, H., and Arguello, H. (2021c). Compressive spectral imaging via virtual side information. *IEEE Transactions on Computational Imaging*, 7:114–123.

Martin, G. and Bioucas-Dias, J. M. (2016). Hyperspectral Blind Reconstruction from Random Spectral Projections. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(6):2390–2399.

Martín, G., Bioucas-Dias, J. M., and Plaza, A. (2015). HYCA: A New Technique for Hyperspectral Compressive Sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5):2819–2831.

Miao, X., Yuan, X., Pu, Y., and Athitsos, V. (2019). l-net: Reconstruct hyperspectral images from a snapshot measurement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Mohammadi, M. R., Fatemizadeh, E., and Mahoor, M. H. (2014). PCA-based dictionary building for accurate facial expression recognition via sparse representation. *Journal of Visual Communication and Image Representation*, 25(5):1082–1092.

Mojica, E., Pertuz, S., and Arguello, H. (2017). High-resolution coded-aperture design for compressive X-ray tomography using low resolution detectors. *Optics Communications*, 404(May):103–109.

Monroy, B., Bacca, J., and Arguello, H. (2022). Jr2net: a joint non-linear representation and recovery network for compressive spectral imaging. *Appl. Opt.*, 61(26):7757–7766.

Monsalve, J., Marquez, M., Esnaola, I., and Arguello, H. (2021). Compressive covariance matrix estimation from a dual-dispersive coded aperture spectral imager. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2823–2827. IEEE.

Monsalve, J., Marquez, M., Sanchez, K., Hinojosa, C., Esnaola, I., and Arguello, H. (2022a). Cocosvi: Single snapshot compressive spectral video via covariance matrix estimation. In *2022*

*12th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–6.

Monsalve, J., Ramirez, J., Esnaola, I., and Arguello, H. (2022b). Covariance estimation from compressive data partitions using a projected gradient-based algorithm. *IEEE Transactions on Image Processing*, 31:4817–4827.

Monsalve, J., Rueda-Chacon, H., and Arguello, H. (2020). Sensing matrix design for compressive spectral imaging via binary principal component analysis. *IEEE Transactions on Image Processing*, 29:4003–4012.

Mueller, A. A., Hausold, A., and Strobl, P. (2002). Hysens-dais/rosis imaging spectrometers at dlr. In *SPIE 4545, Remote Sensing for Environmental Monitoring, GIS Applications, and Geology*, volume 4545, page 11.

Parikh, N. and Boyd, S. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, 1:123–231.

Pinilla, S. E., Vargas García, H. M., and Arguello Fuentes, H. (2016). Probability of correct reconstruction in compressive spectral imaging. *Ingenieria e Investigacion*, 36(2):68–77.

Pourkamali-Anaraki, F. (2016). Estimation of the sample covariance matrix from compressive measurements. *IET Signal Processing*, 10(9):1089–1095.

Qi, H. and Hughes, S. M. (2012). Invariance of principal components under low-dimensional

random projection of the data. *Proceedings - International Conference on Image Processing, ICIP*, pages 937–940.

Quer, G., Masiero, R., Pillonetto, G., Rossi, M., and Zorzi, M. (2012). Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework. *IEEE Transactions on Wireless Communications*, 11(10):3447–3461.

Rabanser, S., Shchur, O., and Günnemann, S. (2017). Introduction to tensor decompositions and their applications in machine learning. *arXiv preprint arXiv:1711.10781*.

Ramirez, J. M., Martínez-Torre, J. I., and Arguello, H. (2021). Ladmm-net: An unrolled deep network for spectral image fusion from compressive data. *Signal Processing*, 189:108239.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv*.

Romero, D., Arianandа, D. D., Tian, Z., and Leus, G. (2016a). Compressive Covariance Sensing: Structure-based compressive sensing beyond sparsity. *IEEE Signal Processing Magazine*, 33(1):78–93.

Romero, D., Arianandа, D. D., Tian, Z., and Leus, G. (2016b). Compressive covariance sensing: Structure-based compressive sensing beyond sparsity. *IEEE Signal Processing Magazine*, 33(1):78–93.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F., editors,

*Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham. Springer International Publishing.

Rubinstein, R., Faktor, T., and Elad, M. (2012). K-svd dictionary-learning for the analysis sparse model. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5405–5408. IEEE.

Rueda, H., Arguello, H., and Arce, G. R. (2016). Compressive spectral testbed imaging system based on thin-film color-patterned filter arrays. *Appl. Opt.*, 55(33):9584–9593.

Serengil, S. I. and Ozpinar, A. (2021). Hyperextended lightface: A facial attribute analysis framework. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pages 1–4.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

Sousa, J. J., Toscano, P., Matese, A., Di Gennaro, S. F., Berton, A., Gatti, M., Poni, S., Pádua, L., Hruška, J., Morais, R., and Peres, E. (2022). Uav-based hyperspectral monitoring using push-broom and snapshot sensors: A multisite assessment for precision viticulture applications. *Sensors*, 22(17).

Steland, A. (2018). Shrinkage for covariance estimation: Asymptotics, confidence intervals, bounds and applications in sensor monitoring and finance. *Statistical Papers*, 54:1441–1462.

Strang, G. (2005). *Linear Algebra and Its Applications*, sec. 6.4, pages 342–344. Thomson Learning, 4 edition.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Tang, Y. (2013). Deep learning using linear support vector machines.

Tao, C., Pan, H., Li, Y., and Zou, Z. (2015). Unsupervised spectral–spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, 12(12):2438–2442.

Testa, M. and Magli, E. (2016). Compressive estimation and imaging based on autoregressive models. *IEEE Transactions on Image Processing*, 25(11):5077–5087.

Tropp, J. A. (2011). User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434.

Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. In Eldar, Y. C. and Kutyniok, G., editors, *Compressed Sensing*, pages 210–268. Cambridge University Press, Cambridge.

Wang, X., Wang, X., Zhao, K., Zhao, X., and Song, C. (2022). Fsl-unet: Full-scale linked unet with spatial–spectral joint perceptual attention for hyperspectral and multispectral image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14.

Wright, S., Nowak, R., and Figueiredo, M. (2009). Sparse Reconstruction by Separable Approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493.

Xanthopoulos, P., Pardalos, P. M., Trafalis, T. B., Xanthopoulos, P., Pardalos, P. M., and Trafalis, T. B. (2013). Linear discriminant analysis. *Robust data mining*, pages 27–33.

Xiao, J., Li, J., Yuan, Q., and Zhang, L. (2022). A dual-unet with multistage details injection for hyperspectral image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13.

Xiong, Z., Xie, A., Sun, D.-W., Zeng, X.-A., and Liu, D. (2015). Applications of Hyperspectral Imaging in Chicken Meat Safety and Quality Detection and Evaluation: A Review. *Critical Reviews in Food Science and Nutrition*, 55(9):1287–1301.

Xu, Q., Zhang, C., and Zhang, L. (2015). Denoising convolutional neural network. In *2015 IEEE International Conference on Information and Automation*, pages 1184–1187.

Yuan, X., Liu, Y., Suo, J., and Dai, Q. (2020). Plug-and-play algorithms for large-scale snapshot compressive imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1447–1457.

Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L. (2019). S4l: Self-supervised semi-supervised

learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1476–1485.

Zhang, S., Huang, H., and Fu, Y. (2019). Fast parallel implementation of dual-camera compressive hyperspectral imaging system. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(11):3404–3414.

Zhang, Y., D'Aspremont, A., and El Ghaoui, L. (2012). Sparse PCA: Convex relaxations, algorithms and applications. *International Series in Operations Research and Management Science*, 166:915–940.

Zhao, C., Zhang, J., Wang, R., and Gao, W. (2018). Cream: Cnn-regularized admm framework for compressive-sensed image reconstruction. *IEEE Access*, 6:76838–76853.

Zhou, P., Han, J., Cheng, G., and Zhang, B. (2019). Learning compact and discriminative stacked autoencoder for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 57(7):4823–4833.

Zhu, F., Wang, Y., Xiang, S., Fan, B., and Pan, C. (2014). Structured sparse method for hyperspectral unmixing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88:101–118.

# Appendices

## Appendix . Appendix for Sensing Matrix Design using the Signal Covariance Matrix

### Deflate Covariance Matrix

In Algorithm 2.1 the covariance matrix $\mathbf{\Sigma}$ is deflated by using the expression

$$\mathbf{\Sigma}_{j+1} \leftarrow \mathbf{\Sigma}_j - \mathbf{\Sigma}_j\mathbf{P} - \mathbf{P}\mathbf{\Sigma}_j + \mathbf{P}\mathbf{\Sigma}_j\mathbf{P}, \tag{96}$$

where $\mathbf{P} = \mathbf{q}\mathbf{q}^T/||\mathbf{q}||^2$ is a symmetric matrix. To proof that the influence of $\mathbf{q}$ in $\mathbf{\Sigma}$ is removed in (96), first define $\mathbf{P}^\perp = (\mathbf{I}-\mathbf{P})$ as the projection matrix for the orthogonal complement of $span(\mathbf{P})$. Thus, the orthogonal projection of a set of vectors $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_n]$ onto the subspace spanned by $\mathbf{P}^\perp$ is given by

$$\mathbf{F}_{P^\perp} = (\mathbf{I}-\mathbf{P})\mathbf{F}. \tag{97}$$

Then, the covariance matrix of the projected data, which can be estimated as

$$\mathbf{\Sigma}_{P^\perp} = \frac{1}{n}\mathbf{F}_{P^\perp}\mathbf{F}_{P^\perp}^T = \frac{1}{n}(\mathbf{I}-\mathbf{P})\mathbf{F}\mathbf{F}^T(\mathbf{I}-\mathbf{P}) \tag{98}$$

yields to

$$\mathbf{\Sigma}_{P^\perp} = (\mathbf{I}-\mathbf{P})\mathbf{\Sigma}(\mathbf{I}-\mathbf{P}) = \mathbf{\Sigma} - \mathbf{\Sigma}\mathbf{P} - \mathbf{P}\mathbf{\Sigma} + \mathbf{P}\mathbf{\Sigma}\mathbf{P}, \tag{99}$$

after replacing $\mathbf{\Sigma} = \mathbf{F}\mathbf{F}^T/n$. Thus, (99) is a closed-form for the covariance matrix of the orthogonal projection of the data. Additionally, it is easy to see that $\mathbf{q}^T\mathbf{\Sigma}_{P^\perp}\mathbf{q} = \mathbf{q}^T\mathbf{\Sigma}\mathbf{q} - \mathbf{q}^T\mathbf{\Sigma}\mathbf{q} - \mathbf{q}^T\mathbf{\Sigma}\mathbf{q} + \mathbf{q}^T\mathbf{\Sigma}\mathbf{q} = 0$.

**Complexity of Algorithm 2.1**

The computational complexity involved in the calculation of the product $c = \frac{\mathbf{q}_j^T}{||\mathbf{q}_j||}\mathbf{\Sigma}\frac{\mathbf{q}_j}{||\mathbf{q}_j||}$, in Line 8 of Algorithm 2.1 has time-complexity of $O(l^2)$. To reduce its complexity, the result from the previous iteration is taken into account. Specifically, define $\mathbf{q}_j = \mathbf{q}_j^{1:k-1} + \mathbf{q}_j^k$ as the vector in the $k^{th}$ iteration, where $\mathbf{q}_j$ has $k$ ones, $\mathbf{q}_j^{1:k-1}$ has $k-1$ ones in the positions selected in the previous $k-1$ iterations, and $\mathbf{q}_j^k$ has a single one in a position being evaluated. Then, in the $k^{th}$ iteration, this product can be calculated as

$$c_k = (\mathbf{q}_j^{1:k-1} + \mathbf{q}_j^k)^T\mathbf{\Sigma}(\mathbf{q}_j^{1:k-1} + \mathbf{q}_j^k) = c_{k-1} + 2(\mathbf{q}_j^{1:k-1})^T\mathbf{\Sigma}\mathbf{q}_j^k + (\mathbf{q}_j^k)^T\mathbf{\Sigma}\mathbf{q}_j^k. \tag{100}$$

Let $\mathbf{q}_j^{1:k-1} = [q_1, q_2, \cdots, q_l]$, $\mathbf{q}_j^k = [0, 0, \cdots, e_k, \cdots, 0, 0]$ with $e_k = 1$ and $\Sigma_j^r$ the elements of the matrix $\mathbf{\Sigma}$. Then, the product $\mathbf{q}_j^T\mathbf{\Sigma}\mathbf{q}_j$ can be written as

$$\mathbf{q}_j^T\mathbf{\Sigma}\mathbf{q}_j = \sum_{r=1}^{l} e_r \sum_{j=1}^{l} q_j\Sigma_j^r = \sum_{j=1}^{l} q_j\Sigma_j^k, \tag{101}$$

where the equality comes from the fact that, the term on the left-side is different from zero only when $r = k$. This latter product demands time-complexity $O(l)$. Finally, the value of $c$ is computed as $c = (1/k)c_k$, since $c_k = \mathbf{q}_j^T\mathbf{\Sigma}\mathbf{q}_j$ and the vector $\mathbf{q}_j$ has not been normalized. Additionally, note

that the product in (101) is computed inside three for-loops, the first one iterates $l$ times in the worst case, the second one performs $l/\tilde{m}$ iterations and the last one spends $l$ iterations; resulting in a computational complexity of $O(l^3)$.

**Convergence of Algorithm 2.1**

Note that the cost function $f(\mathbf{q}) = \mathbf{q}^T\boldsymbol{\Sigma}\mathbf{q}$, defined in (36), is in the feasible set of

$$\lambda_{max}(\boldsymbol{\Sigma}) = \max_{\mathbf{w}\in\mathbb{R}^l, ||\mathbf{w}||=1} \mathbf{w}_i^T\boldsymbol{\Sigma}\mathbf{w}_i \tag{102}$$

which is bounded (Strang, 2005). Additionally, since Algorithm 2.1 is a greedy-search-based strategy, $c^k > c^{k-1}$, where $c^k$ is the value of the function $f(\mathbf{q})$ at the $k^{th}$ iteration. Then, since (36) is bounded and it increases in every iteration, it converges to a local optimum. Note that it holds, only if $||\mathbf{q}|| = 1$, hence the restriction $q_j^k \in \{0, 1/\sqrt{b_j}\}$ is required.

**Proof of theorem 1: Restricted Isometry Property for the Eigenvectors**

Let $\mathbf{f} = [f_1, f_2, \cdots, f_l]^T \in \mathbb{R}^l$ be an arbitrary pixel of the image such that $\mathbf{f} = \mathbf{W}\boldsymbol{\theta}$, with $\mathbf{W}$ being an orthonormal matrix with the eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$ as columns and $\boldsymbol{\theta}$ a sparse vector with the PCA coefficients. Without loss of generality, assuming $\mathbb{E}(\mathbf{f}) = \mathbf{0}$,

$$\begin{aligned}\mathbb{E}\left(||\boldsymbol{\theta}||_2^2\right) = \mathbb{E}\left(\mathbf{f}^T\mathbf{W}\mathbf{W}^T\mathbf{f}\right) = \mathbb{E}\left(\mathbf{f}^T\mathbf{f}\right) = \mathbb{E}\left(\sum_{r=1}^{l} f_r^2\right)\\ = \sum_{r=1}^{l}\mathbb{E}\left(f_r^2\right) = \sum_{r=1}^{l}\text{var}\left(f_r\right) = \text{trace}\left(\boldsymbol{\Sigma}\right) = \sum_{r=1}^{l}\lambda_r,\end{aligned} \tag{103}$$

with $\boldsymbol{\Sigma}$ being the covariance matrix of the signal, $\mathbb{E}(\cdot)$ the expected value, and $\lambda_r$ an eigenvalue such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq \cdots \geq \lambda_l \geq 0$. Recall that the RIP measures the behavior of a matrix

as an orthonormal system, when sparse combinations are used (Candes and Tao, 2005). Thus, let $\mathbf{W}_m \in \mathbb{R}^{l\times m}$ be a matrix composed by $m$ eigenvectors associated with the $m$ largest eigenvalues of the covariance matrix. If $\mathbf{W}_m = [\mathbf{w}_1, \cdots, \mathbf{w}_m]$ is used as the sensing matrix, then

$$\mathbf{y} = \mathbf{W}_m^T \mathbf{f} = \mathbf{W}_m^T \mathbf{W} \boldsymbol{\theta} = \mathbf{A}\boldsymbol{\theta} \tag{104}$$

with $\mathbf{A} = \mathbf{W}_m^T \mathbf{W} = [\mathbf{I}, \mathbf{0}] \in \mathbb{R}^{m\times l}$. Additionally, define $L = \{1, \cdots, l\}$ as the set of indices for the columns of the sensing matrix $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_l]$. Then, let $\mathbf{A}_P$ be a sub-matrix of $\mathbf{A}$ with columns described by indices $P = P_1 \cup P_2 \subset L$ with $P_1 \subseteq \{1, \cdots, m\}$ and $P_2 \subseteq \{m+1, \cdots, l\}$, cardinality $|P| \leq m$, and $\boldsymbol{\theta}_P$ a vector with coefficients in the positions given by $P$. The norm of the measurements is then calculated as

$$\begin{aligned}\mathbb{E}(||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2) = \mathbb{E}(||\boldsymbol{\theta}_{P_1}||_2^2) = \mathbb{E}(||\mathbf{W}_{P_1}^T\mathbf{f}||_2^2) = \mathbb{E}(\mathbf{f}^T\mathbf{W}_{P_1}\mathbf{W}_{P_1}^T\mathbf{f}) = \mathbb{E}(\sum_{i\in P_1}(\mathbf{w}_i^T\mathbf{f})^2) \\ = \sum_{i\in P_1}\mathbb{E}((\mathbf{w}_i^T\mathbf{f})^2) = \sum_{i\in P_1}\text{var}(\mathbf{w}_i^T\mathbf{f}) = \text{trace}(\mathbf{W}_{P_1}^T\boldsymbol{\Sigma}\mathbf{W}_{P_1}) = \text{trace}(\boldsymbol{\Lambda}_{P1}) = \sum_{i\in P_1}\lambda_i.\end{aligned} \tag{105}$$

Note that $\mathbf{A}_P$ is a subset of columns of a identity matrix, $\boldsymbol{\Sigma} = \mathbf{W}\boldsymbol{\Lambda}\mathbf{W}^T$, and $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalues of the covariance matrix. Further, from (103) it can be concluded that

$$\mathbb{E}(||\boldsymbol{\theta}_P||_2^2) = \sum_{i\in P}\lambda_i = \sum_{i\in P_1}\lambda_i + \sum_{i'\in P_2}\lambda_{i'}. \tag{106}$$

Then, replacing (106) in (105)

$$\begin{aligned}\mathbb{E}(||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2) &= \mathbb{E}(||\boldsymbol{\theta}_P||_2^2) - \sum_{i'\in P_2}\lambda_{i'} = \mathbb{E}(||\boldsymbol{\theta}_P||_2^2)\Big(1 - \frac{\sum_{i'\in P_2}\lambda_{i'}}{\mathbb{E}(||\boldsymbol{\theta}_P||_2^2)}\Big) \\ &= \mathbb{E}(||\boldsymbol{\theta}_P||_2^2)\Big(1 - \frac{\sum_{i\in P_2}\lambda_i}{\sum_{i\in P_1}\lambda_i + \sum_{i'\in P_2}\lambda_{i'}}\Big).\end{aligned} \tag{107}$$

Taking into account that $\sum_{i'\in P_2}\lambda_{(m+1)} = |P_2|\lambda_{(m+1)} \geq \sum_{i'\in P_2}\lambda_{i'}$, then

$$\mathbb{E}(||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2) \geq \mathbb{E}(||\boldsymbol{\theta}_P||_2^2)\Big(1 - \frac{|P_2|\lambda_{(m+1)}}{\sum_{i\in P_1}\lambda_i + \sum_{i'\in P_2}\lambda_{i'}}\Big). \tag{108}$$

Then, $\delta_m = |P_2|\lambda_{m+1}/(\sum_{i\in P_1}\lambda_i + \sum_{i'\in P_2}\lambda_{i'})$. Since in natural scenes most of the information is usually kept by some of the first eigenvectors, $\sum_{i\in P_1}\lambda_i \gg |P_2|\lambda_{m+1}$, thus $0 < \delta_m \ll 1$, and the result holds.

**Proof of corollary 1.1 Restricted Isometry Property for Arbitrary Matrices**

Recall that the sensing problem is given by $\mathbf{Q}^T\mathbf{f} = \mathbf{Q}^T\mathbf{W}\boldsymbol{\theta} = \mathbf{A}\boldsymbol{\theta}$, where

$$\mathbf{A} = \begin{bmatrix} \mathbf{q}_1^T\mathbf{w}_1 & \mathbf{q}_1^T\mathbf{w}_2 & \dots & \mathbf{q}_1^T\mathbf{w}_l \\ \mathbf{q}_2^T\mathbf{w}_1 & \mathbf{q}_2^T\mathbf{w}_2 & \dots & \mathbf{q}_2^T\mathbf{w}_l \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_m^T\mathbf{w}_1 & \mathbf{q}_m^T\mathbf{w}_2 & \dots & \mathbf{q}_m^T\mathbf{w}_l \end{bmatrix} = \begin{bmatrix} a_1^1 & a_2^1 & a_3^1 & \dots & a_l^1 \\ a_1^2 & a_2^2 & a_3^2 & \dots & a_l^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^m & a_2^m & a_3^m & \dots & a_l^m \end{bmatrix}. \tag{109}$$

Then, the norm is given by

$$\begin{aligned}\mathbb{E}(||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2) &= \mathbb{E}(||\mathbf{Q}^T\mathbf{W}_P\boldsymbol{\theta}_P||_2^2) = \mathbb{E}(\boldsymbol{\theta}_P^T\mathbf{W}_P^T\mathbf{Q}\mathbf{Q}^T\mathbf{W}_P\boldsymbol{\theta}_P) = \mathbb{E}(\sum_{i=1}^{m}(\mathbf{q}_i^T\mathbf{W}_P\boldsymbol{\theta}_P)^2) \\ &= \sum_{i=1}^{m}\mathbb{E}((\mathbf{q}_i^T\mathbf{W}_P\boldsymbol{\theta}_P)^2) = \sum_{i=1}^{m}\text{var}((\mathbf{q}_i^T\mathbf{W}_P\boldsymbol{\theta}_P)) = \text{trace}(\mathbf{Q}^T\mathbf{W}_P\boldsymbol{\Lambda}_P\mathbf{W}_P^T\mathbf{Q}).\end{aligned} \tag{110}$$

By replacing $\mathbf{A} = \mathbf{Q}^T\mathbf{W}_P$, (110) becomes

$$\text{trace}(\mathbf{A}_P\boldsymbol{\Lambda}_P\mathbf{A}_P^T) = \sum_{k\in P}\lambda_k\sum_{i=1}^{m}(a_k^i)^2. \tag{111}$$

From (111) it can be concluded that the norm of the vector changes if the sum of the squared elements of the sensing matrix columns is different from 1. However, since the first eigenvalues concentrate most of the variance, it is enough to guarantee that $\sum_{i=1}^{m}(a_k^i)^2 = 1$ holds just for the first columns. Thus, the problem splits as

$$\text{trace}(\mathbf{A}_P\boldsymbol{\Lambda}_P\mathbf{A}_P^T) = \sum_{k\in P}\lambda_k\sum_{i=1}^{m}(a_k^i)^2 = \sum_{k\in P_1}\lambda_k\sum_{i=1}^{m}(a_k^i)^2 + \sum_{k'\in P_2}\lambda_{k'}\sum_{i=1}^{m}(a_{k'}^i)^2. \tag{112}$$

Then, defining $\beta_k = 1-\sum_{i=1}^{m}(a_k^i)^2$ and $\beta_{k'} = 1-\sum_{i=1}^{m}(a_{k'}^i)^2$, leads to

$$\mathbb{E}(||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2) = \mathbb{E}(||\boldsymbol{\theta}_P||_2^2) - \sum_{k\in P_1}\lambda_k\beta_k - \sum_{k'\in P_2}\lambda_{k'}\beta_{k'} = \mathbb{E}(||\boldsymbol{\theta}_P||_2^2)\Big(1 - \frac{\sum_{k\in P_1}\lambda_k\beta_k + \sum_{k'\in P_2}\lambda_{k'}\beta_{k'}}{\sum_{k\in P_1}\lambda_k + \sum_{k'\in P_2}\lambda_{k'}}\Big). \tag{113}$$

Additionally, note that $(-\sum_{k\in P_1}\lambda_k\beta_k - \sum_{k'\in P_2}\lambda_{k'}\beta_{k'})$ can be either positive or negative. For the case when it is positive, note that (113) becomes

$$\begin{aligned}\mathbb{E}(||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2) &\leq \mathbb{E}(||\boldsymbol{\theta}_P||_2^2) + \sum_{k\in P_1}\lambda_k|\beta_k| + \sum_{k'\in P_2}\lambda_{k'}|\beta'_{k'}| \\ &= \mathbb{E}(||\boldsymbol{\theta}_P||_2^2)\Big(1 + \frac{\sum_{k\in P_1}\lambda_k|\beta| + \sum_{k'\in P_2}\lambda_{k'}|\beta'_{k'}|}{\sum_{k\in P_1}\lambda_k + \sum_{k'\in P_2}\lambda_{k'}}\Big),\end{aligned} \tag{114}$$

for the case when $(-\sum_{k\in P_1}\lambda_k\beta - \sum_{k'\in P_2}\lambda_{k'}\beta'_{k'})$ is negative, (113) becomes

$$\begin{aligned}\mathbb{E}(||\mathbf{A}_P\boldsymbol{\theta}_P||_2^2) &\geq \mathbb{E}(||\boldsymbol{\theta}_P||_2^2) - \sum_{k\in P_1}\lambda_k|\beta_k| - \sum_{k'\in P_2}\lambda_{k'}|\beta'_{k'}| \\ &= \mathbb{E}(||\boldsymbol{\theta}_P||_2^2)\Big(1 - \frac{\sum_{k\in P_1}\lambda_k|\beta_k| + \sum_{k'\in P_2}\lambda_{k'}|\beta'_{k'}|}{\sum_{k\in P_1}\lambda_k + \sum_{k'\in P_2}\lambda_{k'}}\Big)\end{aligned} \tag{115}$$

holds.

Note that (114) and (115) represent both sides of the RIP. Then, the RIP constant $\delta_{\tilde{m}}$ is defined as

$$\delta_{\tilde{m}} = \frac{\sum_{k\in P_1}\lambda_k|\beta_k| + \sum_{k'\in P_2}\lambda_{k'}|\beta'_{k'}|}{\sum_{k\in P_1}\lambda_k + \sum_{k'\in P_2}\lambda_{k'}}. \tag{116}$$

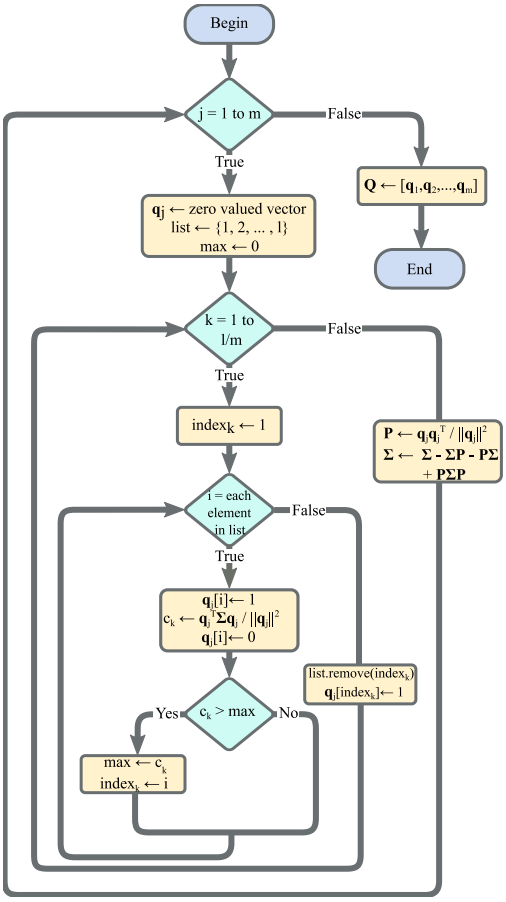


*Figure 45.* Flowchart of Algorithm 2.1. The algorithm begins by initializing a zero-valued vector, then estimates the best position for a one-value by seeking for the position where this entry maximizes the objective function. This step is repeated until placing an additional one-value does not increase the objective function or the transmittance restriction is satisfied.

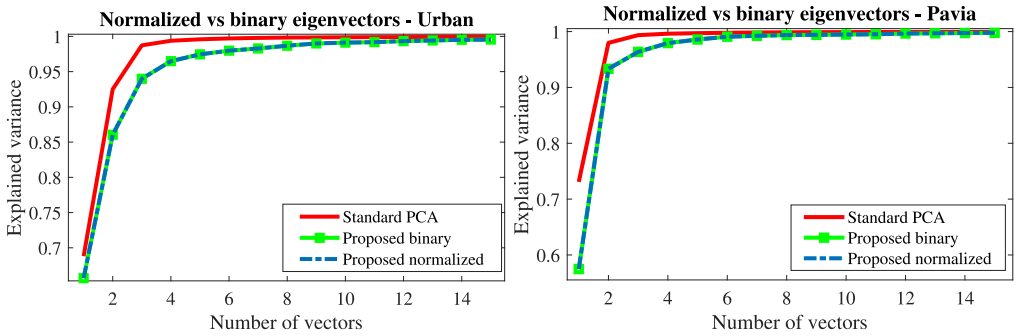


*Figure 46.* Explained variance for binary and discrete (column-normalized) sensing matrices.

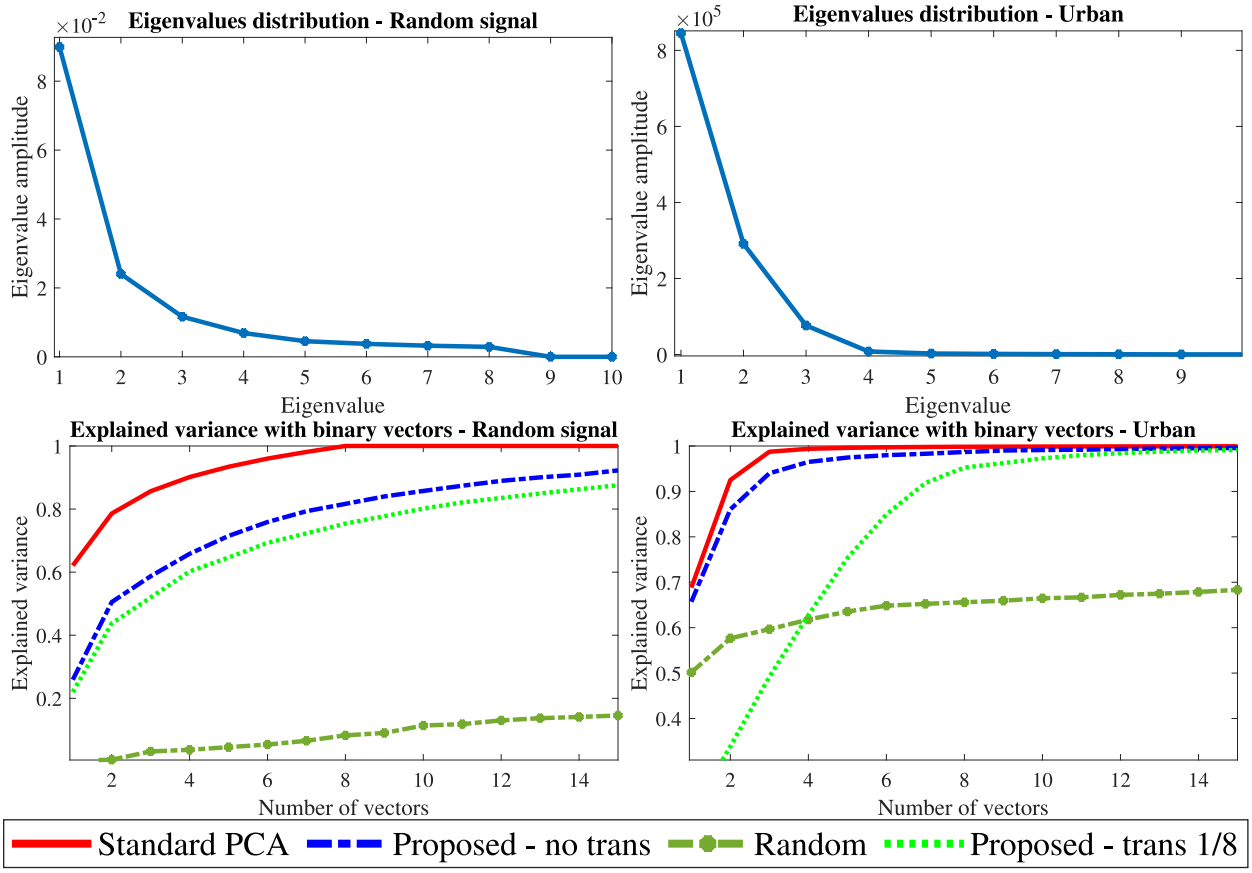


*Figure 47.* Explained variance with different eigenvalue distributions. (Top-left) Eigenvalue distribution for a random signal. (Top-right) Eigenvalue distribution for the Urban dataset. (Bottom-left) Explained variance for a random signal using Standard PCA (red-solid line), proposed without transmittance restriction (dash-dotted blue line), proposed with 1/8 of transmittance (dotted green line), random (dot-dashed green line). (Bottom-right) Explained variance for the Urban dataset.

## Appendix . Mathematical Proofs and Additional Algorithms

**Proof: The expected value of the error term is zero.**

The proposed method assumes the covariance matrix of each subset $\mathbf{X}_i$ is

$$\mathbf{S}_i = \mathbf{S} + \mathbf{R}_i, \tag{117}$$

where $\mathbf{R}_i$ is the error of the subset covariance matrix estimate. Note that

$$\mathbf{S} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j \mathbf{x}_j^T = \frac{1}{n}\Big( \sum_{j_1 \in S_1} \mathbf{x}_{j_1}\mathbf{x}_{j_1}^T + \sum_{j_2 \in S_2} \mathbf{x}_{j_2}\mathbf{x}_{j_2}^T + \cdots + \sum_{j_p \in S_p} \mathbf{x}_{j_p}\mathbf{x}_{j_p}^T \Big), \tag{118}$$

$$= \frac{1}{n}(\mathbf{X}_1\mathbf{X}_1^T + \mathbf{X}_2\mathbf{X}_2^T + \cdots + \mathbf{X}_p\mathbf{X}_p^T)$$

which yields to

$$\mathbf{S} = \frac{1}{p}((\mathbf{S}+\mathbf{R}_1) + (\mathbf{S}+\mathbf{R}_2) + \cdots + (\mathbf{S}+\mathbf{R}_p)) \tag{119}$$

Computing the expectation in both sides yields to

$$\mathbb{E}[\mathbf{S}] = \mathbb{E}[\mathbf{S}] + \frac{1}{p}\mathbb{E}[\mathbf{R}_1 + \mathbf{R}_2 + \cdots + \mathbf{R}_p], \tag{120}$$

which implies that $\mathbb{E}[\mathbf{R}_1 + \mathbf{R}_2 + \cdots + \mathbf{R}_p] = 0$. Additionally, since $\mathbf{R}_i$ is a realization of the random variable $\mathbf{R}$ and using the linearity independence of the expectation we have

$$0 = \mathbb{E}[\mathbf{R}_1 + \mathbf{R}_2 + \cdots + \mathbf{R}_p] = [\mathbb{E}[\mathbf{R}] + \mathbb{E}[\mathbf{R}] + \cdots + \mathbb{E}[\mathbf{R}]] = p\mathbb{E}[\mathbf{R}], \tag{121}$$

which implies that $\mathbb{E}[\mathbf{R}] = 0$.

**Proof of the lemma 3**

Let us define $\mathbf{H}_i \equiv \mathbf{P}_i \mathbf{P}_i^T = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_l]$, and define the matrix

$$\mathbf{B}_i \equiv \mathbf{H}_i \mathbf{R} \mathbf{H}_i, \tag{122}$$

where $\mathbf{H}_i$ is symmetric, and $\mathbb{E}[\mathbf{R}] = 0$ (proof in appendix 6). For simplicity, the $i$ index of matrices $\mathbf{B}$ and $\mathbf{R}$ is dropped. The expected value of the entries of the matrix $\mathbf{B}_{k,j} = \mathbf{h}_k^T \mathbf{R} \mathbf{h}_j$ is given by

$$\mathbb{E}[\mathbf{h}_k^T \mathbf{R} \mathbf{h}_j] = \mathbb{E}[\mathrm{tr}(\mathbf{h}_k^T \mathbf{R} \mathbf{h}_j)] = \quad = \mathrm{tr}(\mathbb{E}[\mathbf{R} \mathbf{h}_j \mathbf{h}_k^T)]. \tag{123}$$

Additionally, the matrix $\mathbf{h}_j \mathbf{h}_k$ is deterministic, and $\mathbb{E}[\mathbf{R}] = 0$, hence

$$\mathbb{E}[\mathbf{h}_k^T \mathbf{R} \mathbf{h}_j] = 0. \tag{124}$$

**Proof of the Cramér-Rao Lower Bound for the estimator**

The variance of the estimator is bounded by

$$\mathrm{var}(\tilde{\mathbf{\Sigma}}) \geq \mathrm{tr}(\mathbf{I}(\mathbf{\Sigma})^{-1}) \tag{125}$$

where $\mathbf{I}(\boldsymbol{\Sigma})$ is the fisher information matrix. To compute $\mathbf{I}(\boldsymbol{\Sigma})$ we observe that

$$\tilde{\mathbf{S}}_i = \frac{n}{p}\mathbf{Y}_i\mathbf{Y}_i^T, \tag{126}$$

with $\frac{n}{p}\tilde{\mathbf{S}}_i \sim \mathscr{W}(\mathbf{P}_i^T\boldsymbol{\Sigma}\mathbf{P}_i + \boldsymbol{\Sigma}_N, n/p)$, and set $r = n/p$ so that that each subset contains $n/p$ samples. The likelihood function is given by

$$f(\tilde{\mathbf{S}}_1,\cdots,\tilde{\mathbf{S}}_p|\boldsymbol{\Sigma}) \propto \prod_{i=1}^{p} \frac{1}{|\boldsymbol{\Sigma}_N + \mathbf{P}_i^T\boldsymbol{\Sigma}\mathbf{P}_i|^{\frac{n}{p}}} \times \left|\mathbf{Y}_i\mathbf{Y}_i^T\right|^{\frac{n}{p}-m} \times \text{etr}\left\{-(\boldsymbol{\Sigma}_N + \mathbf{P}_i^T\boldsymbol{\Sigma}\mathbf{P}_i)^{-1}(\mathbf{Y}_i\mathbf{Y}_i^T)\right\}, \tag{127}$$

where $|.|$ stands for the determinant and etr$\{.\}$ the exponential of the trace. Applying the logarithm in both sides yields

$$\begin{aligned} F(\tilde{\mathbf{S}}_1,\cdots,\tilde{\mathbf{S}}_p|\boldsymbol{\Sigma}) \propto{}& \frac{-n}{p}\sum_{i=1}^{p}\log|\boldsymbol{\Sigma}_N + \mathbf{P}_i^T\boldsymbol{\Sigma}\mathbf{P}_i| + \\ &(\frac{n}{p}-m)\sum_{i=1}^{p}\log|\mathbf{Y}_i\mathbf{Y}_i^T| - \sum_{i=1}^{p}\text{tr}\left((\boldsymbol{\Sigma}_N + \mathbf{P}_i^T\boldsymbol{\Sigma}\mathbf{P}_i)^{-1}(\mathbf{Y}_i\mathbf{Y}_i^T)\right), \end{aligned} \tag{128}$$

where $F(\tilde{\mathbf{S}}_1,\cdots,\tilde{\mathbf{S}}_p|\boldsymbol{\Sigma}) = \log f(\tilde{\mathbf{S}}_1,\cdots,\tilde{\mathbf{S}}_p|\boldsymbol{\Sigma})$. Differentiating twice with respect to $\boldsymbol{\sigma} = \text{vec}(\boldsymbol{\Sigma})$ yields

$$\frac{\partial^2 F(\tilde{\mathbf{S}}_i|\boldsymbol{\Sigma})}{\partial\boldsymbol{\sigma}\partial\boldsymbol{\sigma}^T} = \frac{n}{p}\sum_{i=1}^{p}\mathbf{P}_i\mathbf{A}_i^T\mathbf{P}_i^T \otimes \mathbf{P}_i\mathbf{A}_i\mathbf{P}_i^T + \mathbf{P}_i\mathbf{A}_i^T\tilde{\mathbf{S}}\mathbf{A}_i^T\mathbf{P}_i^T \otimes \mathbf{P}_i\mathbf{A}_i\mathbf{P}_i^T - \mathbf{P}_i\mathbf{A}_i^T\mathbf{P}_i^T \otimes \mathbf{P}_i\mathbf{A}_i\tilde{\mathbf{S}}\mathbf{A}_i\mathbf{P}_i^T, \tag{129}$$

with $\mathbf{A}_i = (\mathbf{\Sigma}_N + \mathbf{P}_i^T \mathbf{\Sigma} \mathbf{P}_i)^{-1}$. The fisher information matrix is computed by calculating the expectation of (129) which yields to

$$\mathbf{I}(\mathbf{\Sigma}) = \frac{n}{p} \sum_{i=1}^{p} \mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T. \tag{130}$$

Note that, from (129) to (130) we used $\mathbb{E}\left[\mathbf{P}_i \mathbf{A}_i^T \tilde{\mathbf{S}} \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T\right] = \frac{n}{p} \mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T$, since $\mathbf{P}_i, \mathbf{A}_i$ are deterministic matrices and $\mathbb{E}\left[\tilde{\mathbf{S}}\right] = \frac{n}{p}(\mathbf{\Sigma}_N + \mathbf{P}_i^T \mathbf{\Sigma} \mathbf{P}_i)$. Hence, the estimator variance of $\mathbf{\Sigma}$ is bounded by

$$\text{var}(\tilde{\mathbf{\Sigma}}) \geq \frac{p}{n} \text{Tr} \left( \sum_{i=1}^{p} \mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T \right)^{-1}. \tag{131}$$

**Proof lemma 2**

Given matrices $\mathbf{P}_i \in \mathbb{R}^{l \times m}$ and $\mathbf{A}_i \in \mathbb{R}^{m \times m}$ with $m <= l$ it holds that

$$\text{rank}(\mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T) = \text{rank}(\mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T) \leq m, \tag{132}$$

using the fact that $\text{rank}(\mathbf{C} \otimes \mathbf{D}) = \text{rank}(\mathbf{C})\text{rank}(\mathbf{D})$ and $\text{rank}(\mathbf{C} + \mathbf{D}) \leq \text{rank}(\mathbf{C}) + \text{rank}(\mathbf{D})$, it can be concluded that

$$\text{rank} \left( \sum_{i=1}^{p} \mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T \right) \leq m^2 p. \tag{133}$$

Hence, for fixed values of $m$ and $l$ the matrix

$$\mathbf{I}(\mathbf{\Sigma}) = \sum_{i=1}^{p} \mathbf{P}_i \mathbf{A}_i^T \mathbf{P}_i^T \otimes \mathbf{P}_i \mathbf{A}_i \mathbf{P}_i^T \in \mathbb{R}^{l^2 \times l^2}, \tag{134}$$

is singular if $p < l^2/m^2$.

**Filtering reduces the variance of the error term**

Note that, note that $\mathbb{E}_B(\hat{f}(\mathbf{\Sigma})_{ij}) = f(\mathbf{\Sigma})_{ij}$ with $\sigma_f = \text{var}(f(\mathbf{\Sigma})_{ij}) = \text{var}(\mathbf{B}_{i,j})$ assuming that the entries of $\mathbf{B}$ are i.i.d.; then averaging in $k \times k$ window yields to

$$\mathbb{E}_B\left[\mathbf{K} * \nabla\hat{f}(\mathbf{\Sigma})\right]_{i,j} = \frac{1}{k^2}\mathbb{E}\left[\sum_{\iota,\rho=-k/2}^{k/2} \nabla\hat{f}(\mathbf{\Sigma})_{i+\iota,j+\rho}\right], \tag{135}$$

with $\mathbb{E}_B$ the expectation over $\mathbf{B}$. Replacing (73) into (135)

$$\begin{aligned}\mathbb{E}\left[\mathbf{K} * \nabla\hat{f}(\mathbf{\Sigma})\right]_{i,j} &= \frac{1}{k^2}\sum_{\iota,\rho}\mathbb{E}\left[\nabla f(\mathbf{\Sigma})_{i+\iota,j+\rho} + \mathbf{B}_{i+\iota,j-\rho}\right] = \frac{1}{k^2}\sum_{\iota,\rho}\nabla f(\mathbf{\Sigma})_{i+\iota,j+\rho}.\\ \left[\mathbf{K} * \nabla\hat{f}(\mathbf{\Sigma})\right]_{ij} &= \frac{1}{k^2}\sum_{\iota,\rho}\nabla f(\mathbf{\Sigma})_{i+\iota,j+\rho} + \mathbf{B}_{i+\iota,j-\rho}\end{aligned} \tag{136}$$

The variance is

$$\mathbb{E}_B\left[\left\{\left(\mathbf{K} * \nabla\hat{f}(\mathbf{\Sigma})\right)_{i,j} - \mathbb{E}_B\left[\mathbf{K} * \nabla\hat{f}(\mathbf{\Sigma})\right]_{i,j}\right\}^2\right] = \mathbb{E}\left[\left(\frac{1}{k^2}\sum_{\iota,\rho}\mathbf{B}_{i+\iota,j-\rho}\right)^2\right] = \frac{\sigma_f}{k^2}. \tag{137}$$

**Bound of the norm for the error term**

Defining $\mathbf{H}_i = \mathbf{P}_i\mathbf{P}_i^T$, the $\ell_2$ norm of the error term for a fixed $i$ in lemma (1) is given by

$$\|\mathbf{H}_i\mathbf{R}_i\mathbf{H}_i\|_2 \leq \sigma_{max}(\mathbf{H}_i)\|\mathbf{R}_i\mathbf{H}_i\|_2 \leq \sigma_{max}(\mathbf{H}_i)^2\|\mathbf{R}_i\|_2. \tag{138}$$

From (10) we have that with probability at least $1-2\exp(-t^2 l)$ it holds that $\|\mathbf{R}_i\| \leq \varepsilon$. Using this fact, with high probability it holds

$$\|\mathbf{H}_i\mathbf{R}_i\mathbf{H}_i\|_2 \leq \sigma_{max}(\mathbf{H}_i)^2\varepsilon \leq \sigma_m^2\varepsilon, \tag{139}$$

with $\sigma_m = max(\sigma_{max}(\mathbf{H}_1)^2, \ldots, \sigma_{max}(\mathbf{H}_p)^2)$, using $||\mathbf{A}\mathbf{A}^H||_2 = ||\mathbf{A}||_2^2$ yields to

$$\|(\mathbf{H}_i\mathbf{R}_i\mathbf{H}_i)^2\|_2 \leq \sigma_m^4\varepsilon^2, \tag{140}$$

Additionally, using the triangle inequality and (139), yields to

$$\left\|\sum_{i=1}^{p}(\mathbf{H}_i\mathbf{R}_i\mathbf{H}_i)^2\right\|_2 \leq p\sigma_m^4\varepsilon^2 = p\sigma_H, \tag{141}$$

hence, the non-commutative Bernstein-type inequality(Tropp, 2011) establishes that for all $t \geq 0$

$$\mathbb{P}\left\{\left\|\sum_{i=1}^{p}\mathbf{H}_i\mathbf{R}_i\mathbf{H}_i\right\|_2 \geq t\right\} \leq 2l \times e^{\frac{-t^2/2}{p^2\sigma_H^2+\sigma_m^2\varepsilon t/3}}, \tag{142}$$

with high probability. Equation (141) shows that the error term increases linearly with the number of partitions, but additionally more partitions also increase the error term $\varepsilon$ which increases in a quadratic manner.

**Experimental distribution of the error term**

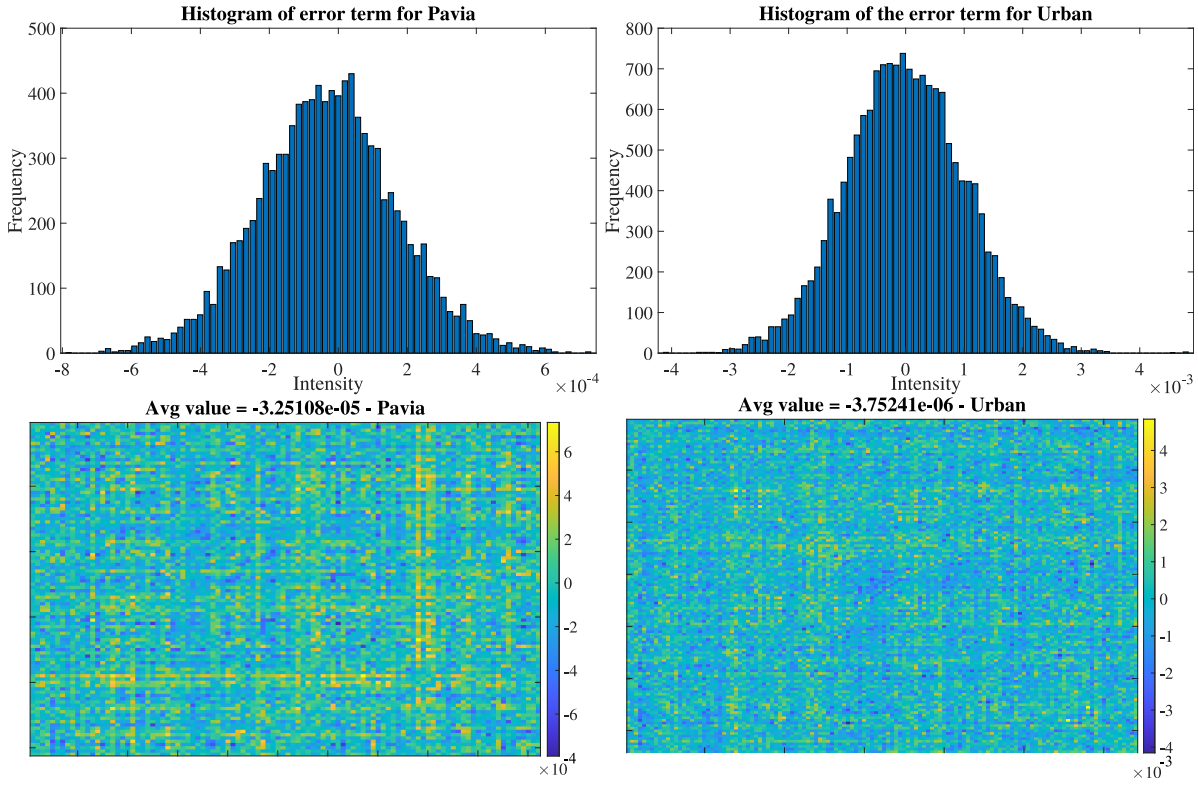


*Figure 48.* Error term of the partition procedure given in (66). Top: Histogram of the error term matrix entries. Bottom: Spatial distribution of the error term.

In order to validate it computationally this term (66) is computed. Figure 48-bottom shows the spatial distribution of the error term for both images. Figure 48-top shows the histogram of the error term values. It can be seen that the values are placed around zero and the sample mean is $-3.25e^{-5}$, $-3.75e^{-6}$ for Pavia and Urban respectively.

**Error term analysis**

Additionally, to test the impact of the error term given in lemma 1, this term is computed and subtracted from the gradient and the results shown in Fig. 49. Results show an important improvement when the filtered gradient is used specially using uniform sensing matrices. Additionally, when the error term is computed (only for comparison purposes) the improvement is up

to one order of magnitude, note that this is only possible if the covariance matrix is known beforehand; in the case of the Uniform matrices the simulations shows no improvement when the error term is subtracted. However, the error always decrease using the filtered gradient and the improvement is larger when using Uniform matrices. Additionally, the stability of the reconstruction appears to improve as well.

**Noise analysis**

Several simulations varying the noise level were performed to test how robust to noise the proposed algorithm is. The Figure 50 shows the NMSE of the reconstructed covariance matrix varying the SNR from 40 dB to 15 dB. It can be seen that the proposed algorithm outperforms all methods when high levels of noise are present in the measurements. For instance, in the Gaussian scenario, the NMSE does not vary that much even though the noise level increases. SpeCA algorithm obtains better results using Gaussian matrices but only for low levels of noise.

**Choosing regularizer parameter.**

The optimal parameter should be chosen as $\tau = \rho \text{trace}(\mathbf{\Sigma})$, where $\rho \in [0,1]$ and depends on the rank of the covariance matrix and variance of the noise. Since $\mathbf{\Sigma}$ is unknown we used the initialization $\mathbf{S}_0$ of the algorithm in order to approximate it as

$$\mathbf{S}_0 = \frac{1}{p}\sum_{i=1}^{p}(\mathbf{P}_i^T)^{\dagger}\tilde{\mathbf{S}}(\mathbf{P}_i)^{\dagger}, \tag{143}$$

where † represents the Moore-Penrose pseudo-inverse. The figure 51 shows the NMSE of the reconstructed covariance matrix by varying the $\rho$ parameter for two levels of noise (30 dB, 20 dB).

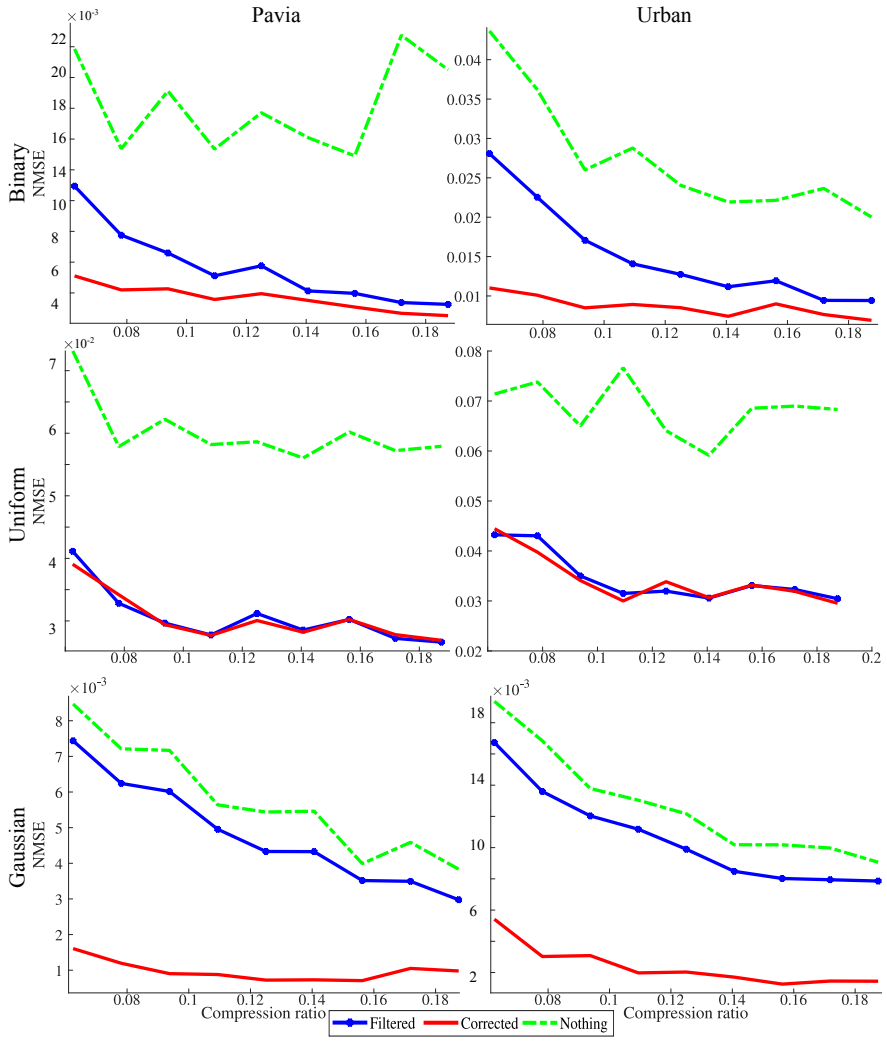


*Figure 49.* NMSE of the recovered covariance matrix when the filtered gradient is used. Dotted green line represents the unfiltered gradient results. Blue solid lines with dot markers show the results with filtered gradient. Red solid line presents the results of filtered gradient when the error term (75) is subtracted in each gradient step.

It can be seen that the

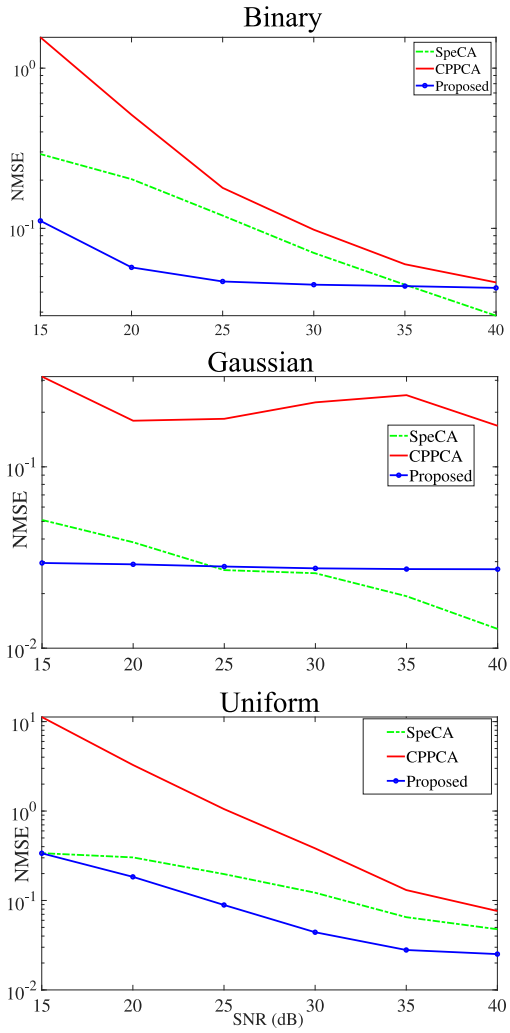


*Figure 50.* NMSE of the reconstructed covariance matrix by varying the noise levels for 8% of compression ratio. Top: Results for Binary sensing matrices. Middle: Results for Gaussian sensing matrices. Bottom: Results for Gaussian Uniform matrices

**Convergence analysis**

This section studies the convergence properties of the projected gradient Algorithm 3.1 to solve (58). Consider the function $\psi(\mathbf{\Sigma}) = \mathrm{Tr}(\mathbf{\Sigma})$ in (58), and let

$$g(\mathbf{\Sigma}) = \sum_{i=1}^{p} ||\mathrm{vec}(\tilde{\mathbf{S}}_i) - \mathbf{Q}_i \mathrm{vec}(\mathbf{\Sigma})||_2^2 + \tau \mathbf{d}^T \mathrm{vec}(\mathbf{\Sigma}), \tag{144}$$

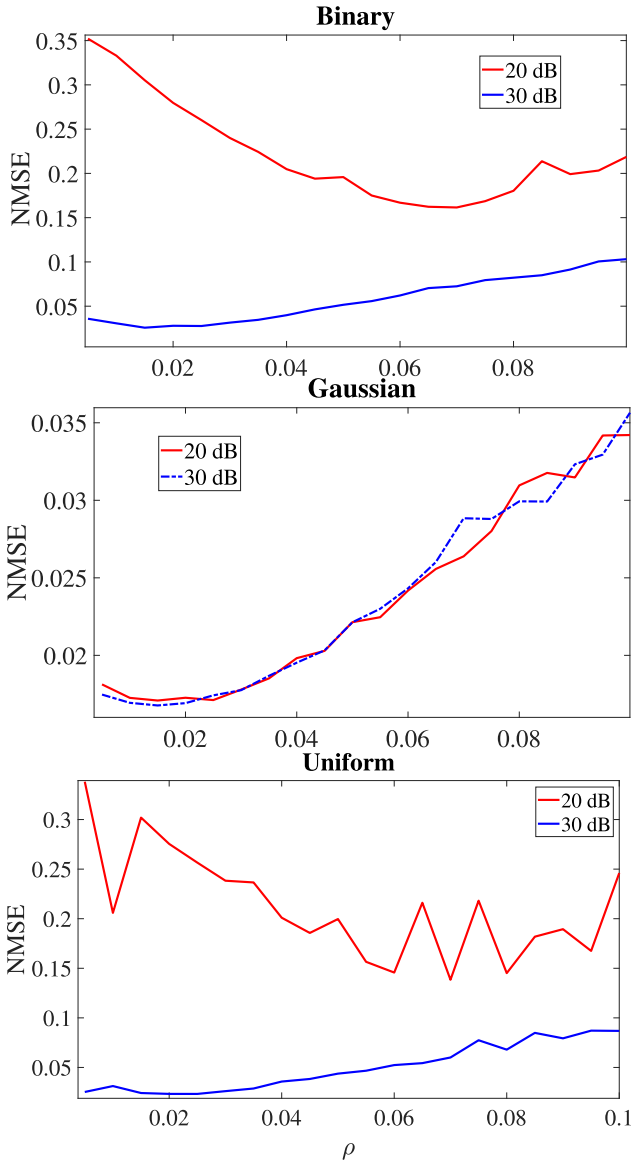


*Figure 51.* NMSE of the reconstruction by varying the $\rho$ parameter. Top: NMSE for Binary matrices. Middle: NMSE for Gaussian matrices. Bottom: NMSE for Uniform matrices

be the vector formulation of (62), with $\mathbf{Q}_i = \mathbf{P}_i^T \otimes \mathbf{P}_i^T$, where $\otimes$ is the Kronecker product, vec$(\cdot)$ denotes the operation that stacks the columns of a given matrix into a column vector, $\mathbf{d} = \text{vec}(\mathbf{I})$ with $\mathbf{I} \in \mathbb{R}^{l \times l}$ the $l \times l$ identity matrix, and $||\cdot||_2$ is the $\ell_2$ norm. Given that $g(\mathbf{\Sigma}) \equiv f(\mathbf{\Sigma})$(Bioucas-

Dias et al., 2014b), the function $g(\boldsymbol{\Sigma})$ is considered instead of $f(\boldsymbol{\Sigma})$,

$$\boldsymbol{\Sigma}^* = \underset{\boldsymbol{\Sigma}\in\mathbb{R}^{l\times l}}{\operatorname{argmin}} \quad g(\boldsymbol{\Sigma}) + h(\boldsymbol{\Sigma}), \tag{145}$$

where $h(\boldsymbol{\Sigma})$ is an indicator function of the positive semi-definitive and Toeplitz set. For simplicity, and taking into account that the function $g(\boldsymbol{\Sigma})$ vectorizes the input matrix, both $g(\boldsymbol{\Sigma})$ and $g(\boldsymbol{\sigma})$ will be used indistinctly, where $\boldsymbol{\sigma} = \text{vec}(\boldsymbol{\Sigma})$ and $\tilde{\mathbf{s}} = \text{vec}(\tilde{\mathbf{S}})$. Further, Fejér proved that sequence of points generated by the projected gradient converges to a solution(Beck, 2017, Theorem 10.23).

**Theorem 3** (Fejér monoticity theorem)**.** Suppose that $g(\boldsymbol{\Sigma})$ and $h(\boldsymbol{\Sigma})$ are proper closed and convex functions, additionally, $\text{dom}(h(\boldsymbol{\Sigma})) \subseteq \text{int}(\text{dom}(g(\boldsymbol{\Sigma})))$ and $g(\boldsymbol{\Sigma})$ is L-*smooth*. Let $\{\boldsymbol{\Sigma}^k\}_{k\geq 0}$ be the sequence of points generated by the projected gradient algorithm. Then for any optimal point $\boldsymbol{\Sigma}^*$ and $k \geq 0$ it holds that

$$||\boldsymbol{\Sigma}^{k+1} - \boldsymbol{\Sigma}^*|| \leq ||\boldsymbol{\Sigma}^k - \boldsymbol{\Sigma}^*||. \tag{146}$$

It can be seen that (145) is convex and differentiable, thus the first assumption of theorem 3 is accomplished. Additionally, a function $g$ is said to be L-smooth if it is differentiable and there exists $L > 0$ such that $||\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})||_2 \leq L||\mathbf{x} - \mathbf{y}||_2$(Beck, 2017), for all $\mathbf{x}, \mathbf{y} \in E$, with $E$ the domain of the function $g$. Thus, for the function $g(\boldsymbol{\Sigma})$ defined in (144), the gradient is given by $\nabla g(\mathbf{x}) = \sum_{i=1}^{p} \mathbf{Q}_i^T (\mathbf{Q}_i\mathbf{x} - \tilde{\mathbf{s}}) + \tau\mathbf{d}$. Plugging it in $||\nabla g(\mathbf{x}) - \nabla g(\mathbf{y})||_2$ yields

$$\left|\left|\left(\sum_{i=1}^{p} \mathbf{Q}_i^T (\mathbf{Q}_i\mathbf{x} - \tilde{\mathbf{s}}_i) + \tau\mathbf{d}\right) - \left(\sum_{i=1}^{p} \mathbf{Q}_i^T (\mathbf{Q}_i\mathbf{y} - \tilde{\mathbf{s}}_i) + \tau\mathbf{d}\right)\right|\right|_2, \tag{147}$$

and after some algebraic manipulations, (147) can be rewritten as

$$\left|\left|\sum_{i=1}^{p}(\mathbf{Q}_i^T\mathbf{Q}_i)(\mathbf{x}-\mathbf{y})\right|\right|_2 \leq \left|\left|\sum_{i=1}^{p}(\mathbf{Q}_i^T\mathbf{Q}_i)\right|\right| \ ||\mathbf{x}-\mathbf{y}||_2. \tag{148}$$

This implies that $L = ||\sum_{i=1}^{p}(\mathbf{Q}_i^T\mathbf{Q}_i)||$. Thus, it can be concluded that the sequence of points generated by Algorithm 3.1 is Fejér monotone which guarantees convergence.

**Incidence of the kernel size of the filtering gradient in the estimation accuracy**

The kernel size in the filtering step affects the reconstruction accuracy since, as shown in Fig. 27, no filtering of the gradient results in poor reconstruction in the eigenvectors associated with the smallest recovered eigenvalues. On the other hand, an over-smoothed gradient can also affect the reconstruction process. To select the correct kernel size $k$ and variance $\sigma$, we perform multiple simulations to show the error versus the variance of the Gaussian kernel. The kernel size was chosen following $k = 2\lceil 2\sigma \rceil + 1$. From Fig. 52, it can be seen that the optimal variance value is $\sigma = 1$.

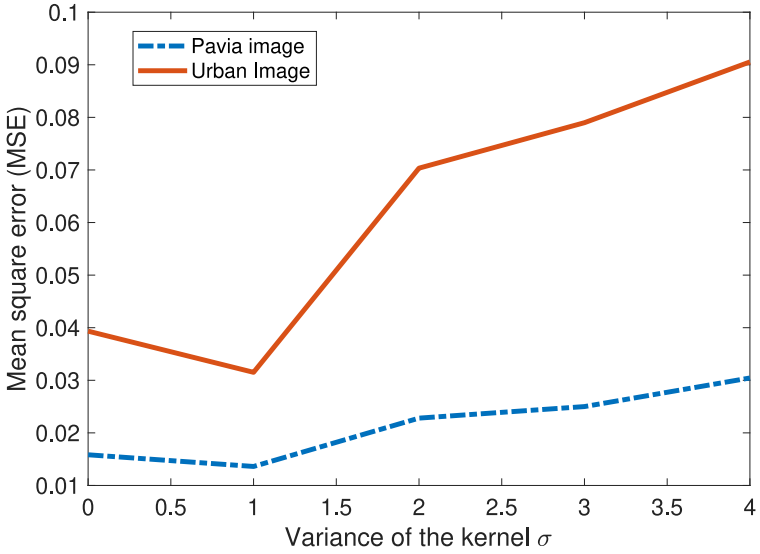


*Figure 52.* Reconstruction error by varying the variance and size of the Gaussian kernel. Note that the kernel size was set to $2\lceil 2\sigma \rceil + 1$.