

https://dspace.org/

A comparative analysis of classic Geometrical methods and sparse regression methods for linearly unmixing hyperspectral image data

Nicolae, Aurel

2019
https://hdl.handle.net/10539/29542

Downloaded from a DSpace repository. https://dspace.org/

A Comparative Analysis of Classic Geometrical Methods and Sparse Regression Methods for Linearly Unmixing Hyperspectral Image Data

Aurel Nicolae

Supervisor: Prof. Michael Sears

Co-supervisor: Dr. Asad Mahmood

July 28, 2019

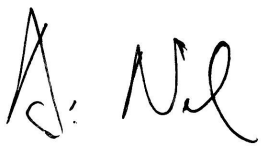
School of Computer Science and Applied Mathematics,
Faculty of Science,
University of the Witwatersrand,
Johannesburg.



A research report submitted to the Faculty of Science, University of Witwatersrand, Johannesburg, in the fulfilment of the requirements for the degree of Masters of Science by Coursework and Research Report.

Declaration

I, Aurel Nicolae (student number: 9703007H), declare that this research report (code: APPM7044A) is my own work. This paper is being submitted for the Degree of Master of Science by Coursework and Research Report, which I am registered for with the University of the Witwatersrand, Johannesburg.
I am aware that plagiarism (the use of someone else's work without their permission or without acknowledging the original source) is wrong. I have followed the required conventions in referencing the thoughts and ideas of others.
I confirm that all of the work submitted for assessment for this qualification is my own unaided work except where I have explicitly indicated otherwise. I declare that I have not submitted this work before for any degree or examination in any other University.



July 28, 2019

Abstract

This research report presents an across-the-board comparative analysis on algorithms for linearly unmixing hyperspectral image data cubes. Convex geometry based endmember extraction algorithms (EEAs) such as the pixel purity index (PPI) algorithm and N-FINDR have been commonly used to derive the material spectral signatures called *endmembers* from the hyperspectral images. The estimation of their corresponding *fractional abundances* is done by solving the related inverse problem in a least squares sense. Semi-supervised sparse regression algorithms such as orthogonal matching pursuit (OMP) and sparse unmixing algorithm via variable splitting and augmented Lagrangian (SUnSAL) bypass the endmember extraction process by employing widely available spectral libraries *a priori*, automatically returning the fractional abundances and sparsity estimates.

The main contribution of this work is to serve as a rich resource on hyperspectral image unmixing, providing end-to-end evaluation of a wide variety of algorithms using different artificial data sets.

Acknowledgements

Thank you to my supervisors Prof. Michael Sears and Dr. Asad Mahmood for their invaluable guidance and advice, and for their expert editing of my work. Mostly, thank you for being patient with me and always being ready to answer my numerous questions.

To my coursework lecturers, thank you to Prof. Joel Moitsheki, Prof. Montaz Ali, Dr. Richard Klein, Dr. Ashleigh Hutchinson, Dr. Terence van Zyl, and especially to Prof. Michael Sears, who deserves another special mention. It was while taking his Digital Image Processing course that inspiration struck me to work on this fascinating research area.

In the loving memory of my mama, Voica Nicolae, who I will always cherish her unconditional love, sense of humour and delicious cooking. Many thanks to my father, Dr. Dan Nicolae, for his love, encouragement and advise.

Last but not least, a very special thank you to my wife, Wendy Nicolae, for believing in me, for her love and moral support, and for making it possible for me to study full-time.

Dedication

To my children, Angelo and Gianna, anything is possible if you put your mind to it and work hard.

Contents

1. Introduction 1
1.1 Research Problem and Motivation 2
1.2 Contribution of this Report 3
1.3 Research Statement and Objective 4
1.4 Delimitations 5
1.5 Research Methodologies 6
1.6 Notation 6
1.7 Report Organisation 6

2. Review of Related Literature - Theoretical Background 8
2.1 Image Spectroscopy 9
2.1.1 Applications 10
2.1.2 Hyperspectral Imaging Processing Techniques 12
2.1.3 Acquisition Modes and Scanners 13
2.1.4 Atmospheric Correction 20
2.1.5 Spatial versus Spectral Resolution 23
2.2 Mathematical Modelling 25
2.2.1 Basic Concepts 25
2.2.2 Assumptions and Conditions 27
2.2.3 Linear Mixing Model Derivation 30
2.3 Hyperspectral Unmixing Processing Chain 34
2.3.1 Dimensionality Reduction - Lower Subspace Projection . . . 36
2.3.2 Affine Projecting onto the Subspace 41
2.3.3 Geometrical Unmixing Methods 42
2.3.4 Pure Pixel Based Algorithms 44
2.3.5 Orthogonal Subspace Projection (OSP) Algorithm 46
2.3.6 Minimum Volume Based Algorithms 47
2.3.7 The Inverse Problem - Estimating Abundance Maps 52
2.4 Sparse Unmixing 55
2.4.1 Norm Regularization 55
2.4.2 Sparse Regression 61
2.4.3 Advantages of Sparse Regression over Classical Approaches . 63
2.4.4 Drawbacks of Sparse Unmixing 64
2.4.5 Sparse Regression Reformulation of the Unmixing Problem . 64
2.4.6 Exact Solutions in the Absence of Noise 66
2.4.7 Approximate Solutions in the Presence of Noise 69

2.4.8 Sparse Unmixing by Variable Splitting and Augmented Lagrangian . . . 70
2.5 Summary of Theoretical Background . . . 72

3. Computer Aided Modeling . . . 73
3.1 Data Simulation . . . 74
3.2 Algorithms . . . 76
3.2.1 Dimension Reduction . . . 76
3.2.2 Geometric Endmember Extraction Algorithms . . . 80
3.2.3 Sparse Unmixing Algorithms . . . 90
3.2.4 ADMM CSR: The SUnSAL Algorithm . . . 93
3.3 Summary of Algorithms . . . 99

4. Experiment Results and Comparative Analysis . . . 100
4.1 Dictionaries used in Simulated Data Experiments . . . 101
4.2 Performance of Endmember Extraction Algorithms . . . 108
4.2.1 Artificially Simulated Toy Data for various SNRs . . . 109
4.2.2 Realistically Simulated Toy Data for various SNRs . . . 113
4.2.3 Artificial Cuprite and Terrain Images . . . 120
4.2.4 USGS vs USGS Pruned Libraries . . . 123
4.2.5 EEAs ComputationTimes . . . 125
4.3 Performance of Unmixing on A Priori Dictionaries v. Image-Extracted Dictionaries . . . 126
4.4 Performance of Reconstructing Various Data . . . 133
4.5 Summary of Results . . . 138

5. Conclusion and Further Research . . . 139
5.1 Conclusion . . . 140
5.2 Future Research . . . 141

Bibliography . . . 142

List of Figures

1.1 Overview of Methodology 7

2.1 Electromagnetic spectrum, Wikipedia. 8
2.2 Hyperspectral imaging concept, [1]. 10
2.3 Applications of Hyperspectral and Multispectral Imaging, [1]. 11
2.4 Multispectral acquisition range, 5 bands example, not drawn to scale, source: GIS Geography 13
2.5 Hyperspectral acquisition range example, not drawn to scale, source: GIS Geography 14
2.6 Push broom scanner, source: Shaw 2003, [2] 16
2.7 Optical signal dispersion, source: Shaw 2003, [2] 17
2.8 Whisk broom scanner, source: RF Wireless 18
2.9 IFOV and GSD 18
2.10 Altitudes and area coverages of various scanners, source: Shaw 2003, [2] 20
2.11 Atmospheric electromagnetic transmittance and opacity, NASA. . . 21
2.12 At-sensor Top-of-Atmosphere reflectance interacting with scatterings, source: Bienarz [3]. 22
2.13 Motivation for hyperspectral data, source: Bioucas-Dias, 2016 Grenoble presentation 23
2.14 Multispectral data cube, source: Yokoya [4] 24
2.15 Hyperspectral data cube, source: Yokoya [4] 24
2.16 Fused data cube, source: Yokoya [4] 24
2.17 Hyperspectral vectors in a lower-dimensional manifold, source: Bioucas-Dias, 2016 Grenoble presentation, modified 25
2.18 Interference from tree tops, neighbouring obstacles, atmosphere, source: [5]. 28
2.19 Linear mixing illustration of measured radiance at a pixel. [5]. . . . 28
2.20 Nonlinear mixing illustration of measured radiance at a pixel. [5]. . . 29
2.21 Convex Hull C for $p = 3$ endmembers [5]. 33
2.22 Hyperspectral Unmixing Processing Chain [5]. 34
2.23 Redudancy levels explained by Shelns [6] for two separate measures r_1 and r_2 37
2.24 PCA explanation given by Shelns [6]. 38
2.25 Illustration of the concept of minimum volume concept [5]. 43
2.26 PPI algorithm projecting data on k-skewers, source: gipsa-lab [7]. . . 44

2.27 N-FINDR algorithm inflates the volume of a convex hull inside data, source: gipsa-lab [7]. 45
2.28 Illustration of the minimum volume simplex concept [5]. 47
2.29 Noisy data with outliers could lead MV to result in wrong estimate, i.e. dashed simplex [5]. 49
2.30 Least Squares representation, source: Bieniarz [3] 55
2.31 Geometric interpretation of ridge regression, source: Bieniarz [3] . . 56
2.32 Geometric interpretation of l_0-norm regularization, source: Bieniarz [3] 57
2.33 Representation of the l_0-"ball" in 3D, which looks like skewers along the coordinate axis. 58
2.34 Geometric interpretation of l_1-norm regularization, source: Bieniarz [3] 59
2.35 (a) The subspaces containing sparse vectors in $\mathbb{R}^3$ lie close to the coordinate axes, which is a nonconvex problem. (b) The l_2-minimization fails to find sparse solution $\boldsymbol{s}$, as the point of contact $\hat{\boldsymbol{s}}$ projected on the green null-space can be anywhere on the red l_2-ball. (c) The l_1-minimizer finds the sparse point-of-contact $\hat{\boldsymbol{s}}$ to be the same as vertex $\boldsymbol{s}$ with high probability, due to the pointiness of the l_1-"ball". [8] . . 60
2.36 Finding which columns in dictionary $\boldsymbol{A}$ are active in the image is a combinatorial problem, source: Bieniarz [3] 62

3.1 Orthogonal projection and DPFT projection, source: Bioucas-Dias [5] 80

4.1 Cuprite image acquired by AVIRIS, 1993. 102
4.2 Terrain image acquired by HYDICE, 1995 102
4.3 (left) DCT coefficients and cummulative energy describing dictionary A_1; (right) A_1's mean signature and a few other randomly chosen signatures. 105
4.4 (left) DCT coefficients and cummulative energy describing dictionary A_2; (right) A_2's mean signature and a few other randomly chosen signatures. 105
4.5 (left) DCT coefficients and cummulative energy describing dictionary A_3; (right) A_3's mean signature and a few other randomly chosen signatures. 106
4.6 (left) DCT coefficients and cummulative energy describing dictionary A_4; (right) A_4's mean signature and a few other randomly chosen signatures. 106
4.7 (left) DCT coefficients and cummulative energy describing dictionary A_5; (right) A_5's mean signature and a few other randomly chosen signatures. 107
4.8 (left) DCT coefficients and cummulative energy describing dictionary A_6; (right) A_6's mean signature and a few other randomly chosen signatures. 107
4.9 Artificial fractional abundances with sparsity $k = 5$. 108
4.10 Realistic fractional abundances with sparsity $k = 5$. 108
4.11 2D Projection of Extracted Endmembers and Artificial Toy Data, $SNR = 20\,dB$. 109

4.12 Endmembers Extracted by Various EEAs for Artificial Toy Data with $SNR = 20\,dB$. 110
4.13 2D Projection of Extracted Endmembers and Artificial Toy Data, $SNR = 50\,dB$. 111
4.14 Endmembers Extracted by Various EEAs for Artificial Toy Data with $SNR = 50\,dB$. 112
4.15 2D Projection of Extracted Endmembers and Realistic Toy Data, $SNR = 20\,dB$. 113
4.16 Endmembers Extracted by Various EEAs for Realistic Toy Data with $SNR = 20\,dB$. 114
4.17 2D Projection of Extracted Endmembers and Realistic Toy Data, $SNR = 50\,dB$. 115
4.18 Artificial fractional abundances with sparsity $k = 5$, instrinsic estimate of $\hat{k} = 5$. 116
4.19 Realistic fractional abundances with sparsity $k = 5$, instrinsic estimate of $\hat{k} = 10$. 116
4.20 Endmembers Extracted by Various EEAs for Realistic Toy Data with $SNR = 50\,dB$. 117
4.21 HySime's mainly underestimates for artificial toy data from library $\boldsymbol{A}_3$. 118
4.22 HySime's estimates for realistic toy data from library $\boldsymbol{A}_3$ tends closer to $2k$. 118
4.23 HySime's mainly underestimates for artificial toy data from library $\boldsymbol{A}_4$. 119
4.24 HySime's mainly underestimates for realistic toy data from library $\boldsymbol{A}_4$. 119
4.25 HySime's very good matches for the Cuprite image $\boldsymbol{A}_5$. 120
4.26 HySime's almost perfect matches for the Terrain image $\boldsymbol{A}_6$. 120
4.27 EEAs performance for artificial toy data from $\boldsymbol{A}_1$. 121
4.28 EEAs performance for realistic toy data from $\boldsymbol{A}_1$. 121
4.29 EEAs performance for artificial toy data from $\boldsymbol{A}_2$. 122
4.30 EEAs performance for realistic toy data from $\boldsymbol{A}_2$. 122
4.31 EEAs performance for artificial toy data from $\boldsymbol{A}_3$. 123
4.32 EEAs performance for realistic toy data from $\boldsymbol{A}_3$. 123
4.33 EEAs performance for artificial toy data from $\boldsymbol{A}_4$. 124
4.34 EEAs performance for realistic toy data from $\boldsymbol{A}_4$. 124
4.35 p_s results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on given dictionaries $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$. 128
4.36 SRE (dB) results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on given dictionaries $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$. 129
4.37 SRE (dB) versus SNR (dB) comparison for various unmixing algorithms on $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$. 129
4.38 p_s results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on image-extracted dictionaries $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$. 130

4.39 SRE (dB) results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on image-extracted mixing matrices $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$. 131
4.40 Random Cuprite pixel 9528 reconstructed spectrum 133
4.41 Random Cuprite pixel 28673 reconstructed spectrum 133
4.42 Random Terrain pixel 620 reconstructed spectrum 134
4.43 Random Terrain pixel 123067 reconstructed spectrum 134
4.44 Comparison of the third endmember's fractional abundance map of the Cuprite image. 134
4.45 Comparison of the reconstructed spectral band 84 of the Cuprite image. 135
4.46 Comparison of the fourth endmember's fractional abundance map of the Terrain image. 135
4.47 Comparison of the reconstructed spectral band 107 of the Terrain image. 136
4.48 p_s plot for the artificial Cuprite and Terrain images at $SNR = 30\,dB$. 136
4.49 SRE plot for the artificial Cuprite and Terrain images at $SNR = 30\,dB$. 137

List of Tables

2.1 Imaging Spectrometers, source: Jose Bioucas-Dias, GISTAM 2015 presentation 14
2.2 Landsat-8 band specifications, source: Landsat-8 15
2.3 Data Cube size calculations based on a data point requiring 16 bits storage in an ENVI file. 16
2.4 The sensor's operational altitude and IFOV are both built into the GSD. 19

4.1 Mutual Coherence and Spark Estimates for Different Libraries . . . 103
4.2 Mean running times for extracting endmembers for artificial data . . 125
4.3 Mean running times unmixing on available *a priori* v. image-extracted dictionaries 132

Chapter 1

Introduction

Remotely sensed hyperspectral imaging, also known as *imaging spectroscopy*, has become a significant research tool in areas such as earth observation, geological studies, to name a few, due to advances in hyperspectral cameras technology, an increased focus on the development of *sparse unmixing* algorithms, and due to widely available spectral libraries from agencies such as the United States Geological Survey (USGS) and the National Aeronautics and Space Administration (NASA) agency.

The volume of remotely sensed hyperspectral data cubes is enormous due to modern hyperspectral sensors's improved spectral resolution now covering several hundred bands spanning the visible and infrared ranges of the electromagnetic spectrum, compared to conventional photographic cameras capturing only three bands in the visible electromagnetic spectrum. The immensity of these "big data" cubes creates great difficulty in storing, transmitting, processing and analysing this data. Hence sparse unmixing, closely related to *compressed sensing*, is absolutely necessary in reducing the large data volumes, and characterising it in terms of a very small number of active material spectral signatures called *endmembers*, thus dealing with the large amount of redundancy.

The hyperspectral data can be visualised as a three-dimensional (3D) cube, which is simply a stack of two-dimensional (2D) images. At each layer the spatially arranged pixel in that 2D image represents the intensity value at a location for a particular spectral band. It is standard practice to reshape the 3D cube into a 2D array with the first dimension as the number of spectral bands and the second as the total number of pixels. Note that we lose the spatial structure after this transformation, however and more importantly, we preserve the realistic abundances critical to identifying materials.

Sparse spectral unmixing is achieved by applying the linear transformation to each pixel's spectral vector:

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha} \tag{1.1}$$

where the observed spectral vector $\boldsymbol{y} \in \mathbb{R}^B$, with B as the number of spectral channels, can be represented as a linear mixture of p endmembers organized as column vectors of the materials matrix $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_p]$. In signal processing, we would have *basis vectors* or *wavelets* instead of the endmembers. These component vectors are also called *atoms* in the literature. Their corresponding coefficients vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_p]^T$ represents the endmembers' contributions in each pixel called *abundance fractions*.

Linearity is often assumed in spectral unmixing. The sparse nature of real-world scenes guarantees that a very small number of contingent materials is present in each pixel, usually 4 or 5 endmembers, [9].

The hyperspectral unmixing process starts with the pre-processing of the raw radiance data cubes, correcting it for atmospheric effects and measurement instrument errors, thus obtaining the reflectance data cubes. The second step of dimensionality reduction is optional, however it has tremendous benefits in terms of computation efficiency. This step entails projecting the observed data onto a lower dimensional subspace. The third step is the classic endmember extraction from the image data based either on convex geometry or Bayesian statistical methods. This step is now bypassed by the use of widely available spectral libraries from USGS, NASA and other agencies. However these libraries are large and contain mutually coherent signatures, thus requiring pruning. Dictionary learning is also an active research area receiving a lot of attention lately, [10], [11], [12], [13], [14], [15].

In the context of this paper, the focus is limited to the geometric endmember extraction algorithms and the spares regression algorithms.

1.1 Research Problem and Motivation

Comparative analysis in published literature either focuses on geometric endmember extraction methods, or on algorithms solving the inverse problem using *sparse regression* approaches. Very few publications present the whole picture, the most notable comparative work has been presented by Jose Bioucas-Dias [5].

The main motive for writing this research report is to present an "across-the-board" analysis of the classic endmember extraction methods, currently still in use

in geological core analysis, and the more recent sparse unmixing methods of estimating fractional abundances. The later methods and advances thereof, such as hyperspectral data fusion, image classification, are becoming more popular in the remote spectroscopy field in Europe and China, [16], [17], [3], [18], [19], [20], [21], [22], [23], [24], [25]. The common practice in the published literature is to test algorithms on simulated data as the "ground truth" is known in advance, i.e. the researcher knows which materials are mixed. Few papers in the literature attempt to verify the algorithms on real data, as for remotely sensed data the ground truth is hard to obtain for every scene. However, the Cuprite scene in Nevada, U.S.A. is commonly used for experimenting on real data, as mineral maps are available on the USGS site.

To demonstrate that the unmixing algorithms work, published researchers not only experiment on simulated hyperspectral data, but they mainly run algorithms on "toy data", i.e. each pixel's reflectance is composed of the same small number of sparse endmembers. This is not very realistic, as Iordache [9] states that the number of materials in a given scene is less than 20, and the number of endmembers present in a mixed pixel are usually 4 or 5. This means no same 5 materials are present in all the pixels, and there usually has to be some spatial correlation between the pixels. Out of say 20 materials in a scene, 4 or 5 different materials will participate in various locations of the image. Thus devising more "realistic" simulated data was the byproduct of this work.

1.2 Contribution of this Report

My main contribution of this work is to serve as rich resource, providing a comprehensive end-to-end comparative analysis of various methodologies and algorithms used to unmix hyperspectral image data. The unmixing process of hyperspectral images is very similar to blind source separation for audio signals, however the higher dimensionality of hyperspectral data makes it more challenging. Unmixing methodology makes plentiful use from various mathematical fields such as linear algebra, geometry, optimization, probability theory, and more recently from machine learning and parallel computing. As computation power is increasing, hypersepectral sensor technology is improving, and data is more abundantly available, so does research on new unmixing methods has to keep growing. As Bioucas-Dias states in [30], publications on Hyperspectral topics have grown comparably to publications on Radar topics. The growing interest in this topic and the combination of mathematical and computational techniques make unmixing hyperspectral images such a fascinating topic.

The secondary contribution, as a byproduct of the main contribution, is to generate realistic artificially simulated data, where say 5 sparse materials are selected at random from the 10 or more endmembers from a library, and to test the unmixing algorithms on these data. This is not to be confused with nonlinear mixtures, as the artificial data is still mixed linearly. This should distribute the number of different materials randomly across the pixels in the simulated scene. In addition, as suggested by my supervisors, I have generated artificial images from real Cuprite images, but selecting only a small number materials in the image, and synthesising the image back thereby eliminating some of the endmembers present. This creates an artificial image where we know in advance the materials present, and the mixed pixels are spatially correlated as they are based on real images.

1.3 Research Statement and Objective

The primary **Research Questions** in this comparative analysis are:

- Which endmember extraction algorithm identifies the active sparse endmembers the best? Is there a tradeoff in performance versus complexity?
- Do available spectral dictionaries perform better than image-extracted dictionaries in reconstructing the image? What is the optimal sparsity of signals which are recoverable using hyperspectral libraries versus image-extracted dictionaries?
- Which sparse unmixing algorithm achieves the most accurate reconstruction on artificial toy data, realistic toy data, and on artificial images generated from real images? What is the probability of success?

Thus based on the research questions the **Problem Statement** aims to address the following concepts:

i Endmember extraction algorithms accuracy;

ii Available spectral dictionaries versus image-extracted dictionaries;

iii Experimental data simulation methods.

The **Research Hypothesis** formulated to predict the outcome of my research questions is as follows:

i Addressing endmember extraction accuarcy, the pixel purity-based algorithms are expected to perform poorly due to pure pixel constraint not likely to hold in most scenarios.

ii Unmixing with spectral dictionaries given *a priori* is expected to outperform image-extracted dictionaries, simply due to endmember extraction algorithms (EEAs) not always identifying all the correct endmembers in a scene.

iii Toy examples are expected to run smoothly due to low complexity. In initial experiments, unmixing artificial images also ran smoothly, but took a longer time to run due to large size. Unmixing struggles somewhat on "non-to" examples to estimate the correct sparsity because the mixing of endmembers is more complex.

The **Research Objectives** for this research report are:

1. Build a framework in MATLAB using existing code already implemented by experts in the field that performs the dimensionality reduction, the endmember extraction algorithms, and sparse unmixing, keeping in mind that this a comparative study.
2. Implementing the proposed simulation methods for toy data, non-toy data, and artificial images.
3. Evaluating and comparing the accuracy of different endmember extraction algorithms using root mean square error (RMSE) metric.
4. Implementing performance measures such as Signal-to-Noise Ratio (SNR), Signal Reconstruction Error (SRE), and probability of success (p_s).
5. Evaluating and comparing the reconstruction of original images on dictionaries vs image-extracted dictionaries.
6. Evaluating and comparing the reconstruction of toy, non-toy and artificial images.

1.4 Delimitations

This report will focus on unmixing artificial images only, as unmixing real images is usually challenging. The reason being that it is difficult to know the ground truth, thus we cannot be certain of the accuracy of our estimations. The Linear Mixing Model will be the basis of this paper, since we can generate the mixtures in artificial images linearly.

1.5 Research Methodologies

Diagram 1.1 illustrates the road map presenting the processes followed. The methodology is a straightforward cascading approach. Once the required library is loaded in MATLAB and the desired parameters are set, we simulate the artificial data sets as per linear mixing model. Various endmember extraction algorithms are then run, compared, and the best performer is chosen. If the original library is used to unmix the data, then there is no further use of the materials matrix extracted from the image. Otherwise, we use the image-extracted materials matrix as the dictionary for unmixing the data. In this step we need to select which algorithm is to be used for unmixing. Then results are analysed to give us the signal-reconstruction error, and other metrics.

1.6 Notation

The convention used is to write scalars in lower-case letters, vectors in bold lower-case letters, matrices in bold upper-case letters. Estimates will usually have a triangular hat above the letter. Norms are represented with double vertical bars.

1.7 Report Organisation

The rest of this research report is organized as follows. Chapter 2 provides a theoretical background on image spectroscopy emphasising the important concepts for spectral unmixing. A brief overview of hyperspectral imaging processing techniques is given next, followed by the all important mathematical modelling ideas. An introduction to linear unmixing is given next, presenting the important stages of the unmixing process flow. Lastly, this chapter discusses the sparse representation of the spectral unmixing problem, offering explanations of l_p norms regularization.

Chapter 3 provides our method of simulating toy hyperspectral data. Next it provides an extensive look at the pseudo-code for the dimension reduction algorithms, endmember extraction algortihms, and sparse regression algorithms, with the main focus on the alternating direction method of Lagrangian multipliers.

In chapter 4 we present experiment results which aim at answering the research questions. Chapter 5 contains the conclusion of this paper and further research recommendations.

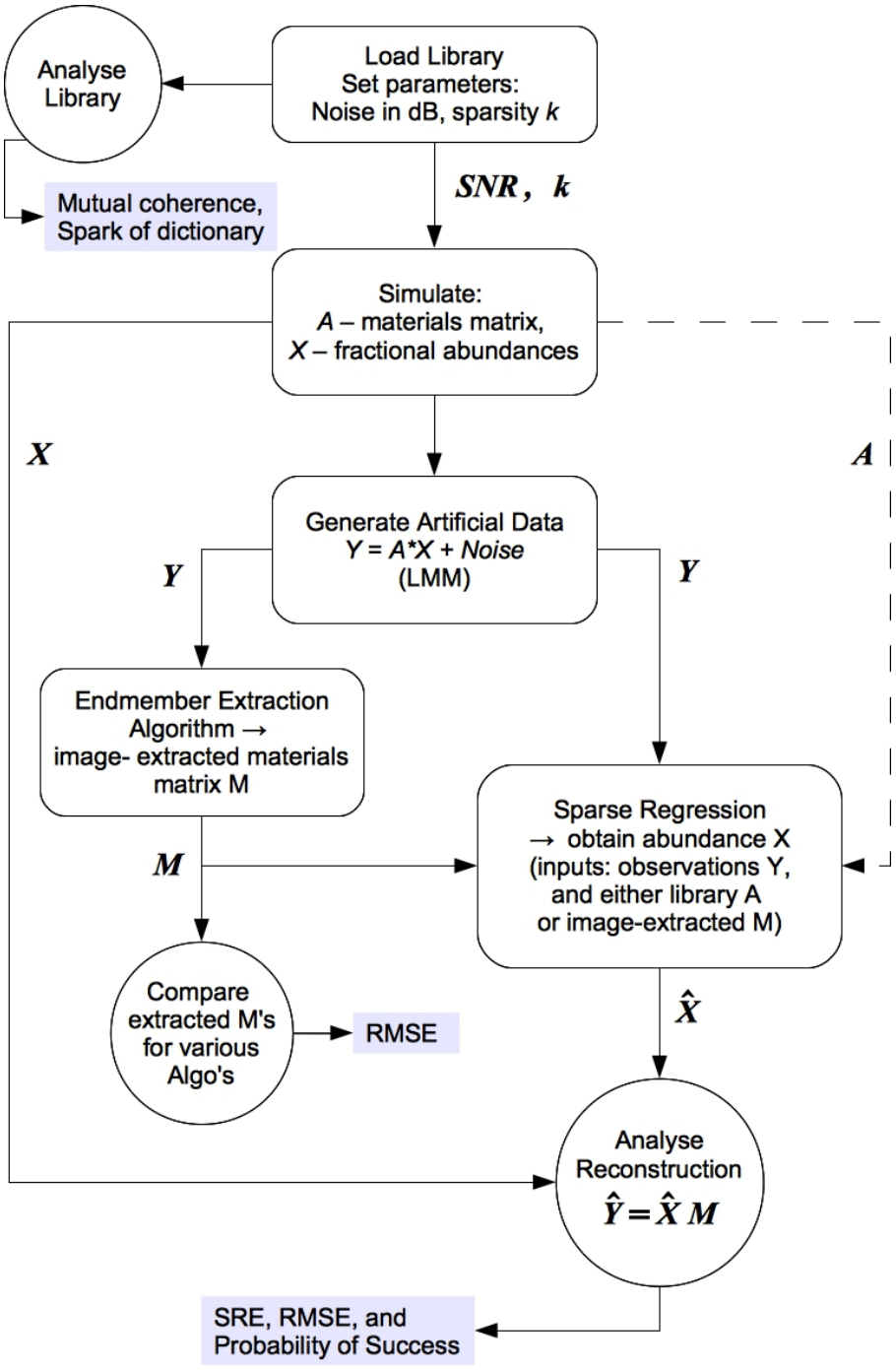


Fig. 1.1: Overview of Methodology

Chapter 2

Review of Related Literature - Theoretical Background

Remote sensing is the science concerned with the measurement, analysis and interpretation of spectra acquired from a given scene, which is observable but not available to direct exploration, [26]. This means that inaccessible scenes must be surveyed from far away distances using airborne or satellite sensors, and mineral rich terrains require core samples to be drilled out and examined in a laboratory.

The human eye's cone cells are sensitive to the "visible light" ranges of wavelengths of the electromagnetic spectrum approximately corresponding to red, green and blue (RGB), roughly between $400nm$ and $700nm$. The three types of cone cells interact with the incoming light reflected off a scene, and filter the RGB components. The brain then synthesizes these three RGB components with the shape information to interpret the scene, [18].

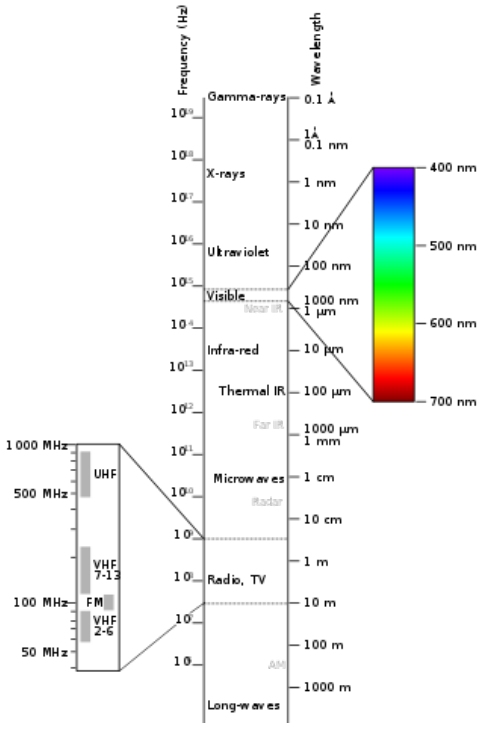


Fig. 2.1: Electromagnetic spectrum, Wikipedia.

Digital cameras also use RGB filters mimicking the human eye, to separate the incoming light on the sensor into red, green and blue components, and directing

them to different charged coupled device (CCD) arrays for storage. The image is then recombined in the computer by synthesizing the RGB components.

Remote sensing allows scientists to "see" what is, otherwise, invisible to the human eye. Hyperspectral and multispectral cameras augment the spectral resolution capabilities of RGB cameras, by incorporating the visible and the near infrared parts of the electromagnetic spectrum. These modern remote sensors acquire images whose pixels represent the spectra of tens to hundreds of spectral bands. This type of high-dimensional imaging is called *Image Spectroscopy*.

2.1 Image Spectroscopy

Image Spectroscopy is concerned with processing hyperspectral and multispectral images, i.e. image processing beyond gray-level images and RGB images. *Hyperspectral Imaging* is defined as the acquisition of images in *hundreds* of contiguous spectral bands such that for each pixel a *radiance spectrum* can be derived, [27]. On the other hand, *Multispectral Imaging* measures light in a small number of spectral bands, typically 3 to 15.

Hyperspectral imaging gives scientists the ability to identify and characterize materials from the finer spectral resolution, which is much more difficult to do from conventional RGB images or the naked human eye. The higher spectral resolution is the reason why hyperspectral images are so popular in image spectroscopy.

Figure 2.2 depicts the measured hyperspecrtal data, which is the signal acquired at each pixel as a mixture of light scattered by the materials present in the field of view of the camera. The recorded radiance responses at all pixels corresponding to a spatial location are organized in planes, each representing a spectral channel forming a hyperspectral data cube.

Slicing the hyperspectral cube at each spectral band, and processing the slice as we would a gray-level image is one way of analyzing this data. However, in hyperspectral imaging, the preferred way is to "drill down" each pixel along all the bands on the spectral axis. This gives us a spectral "signature" for each pixel, which allows us to distinguish between pixels with different spectra.

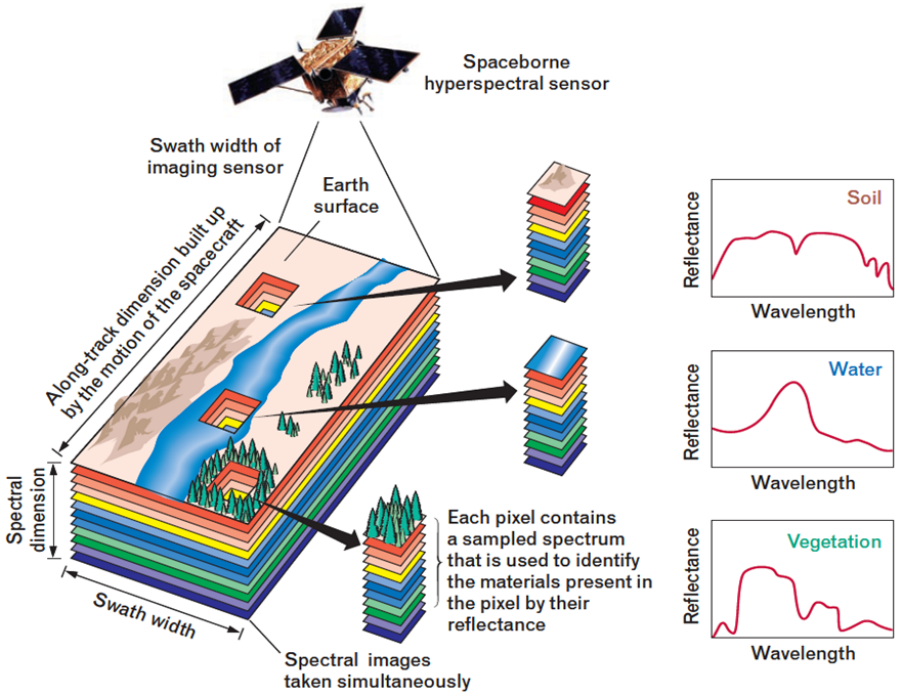


Fig. 2.2: Hyperspectral imaging concept, [1].

If we assume "pure pixels" as the idealistic image 2.2, we would be able to distinguish the "spectral signatures" of materials such as soil, water and vegetation, as each material will respond differently in the visible and near infrared spectrum. This motivates the usefulness of hyperspectral imaging.

2.1.1 Applications

The applications of hyperspectral imaging in remote sensing have a rich range of uses in many industries, [5], [18], such as the following:

- earth observation - monitoring natural landscapes, evaluating the damages of natural disasters, e.g. earthquakes, tornadoes, wild fires and floods;
- man-induced hazards observation - damage evaluation of oil spills and chemical contamination
- agricultural processes - crop monitoring, identifying overripe or damaged fruit;
- military surveillance - observing unaccessible remote areas, identifying defence threats;

- urban development - identifying resources available in an area considered for urban planning;
- planetary exploration - satellites monitoring Mars have hyperspectral sensors;
- geological core examination - identifying chemical compositions of drilled core;
- and many more, such as climate change monitoring, ecology, forestry, marine, astronomy, etc.

Image spectroscopy is also applied at laboratory scale as well as in forensics, pathology, biometrics, biomedical control, medical diagnostics, pharmaceutical quality control, food safety and quality control, etc.
The main advantage of hyperspectral imaging is that high spectral resolution allows for detection, identification and quantification of surface materials, as well as inferring material properties.

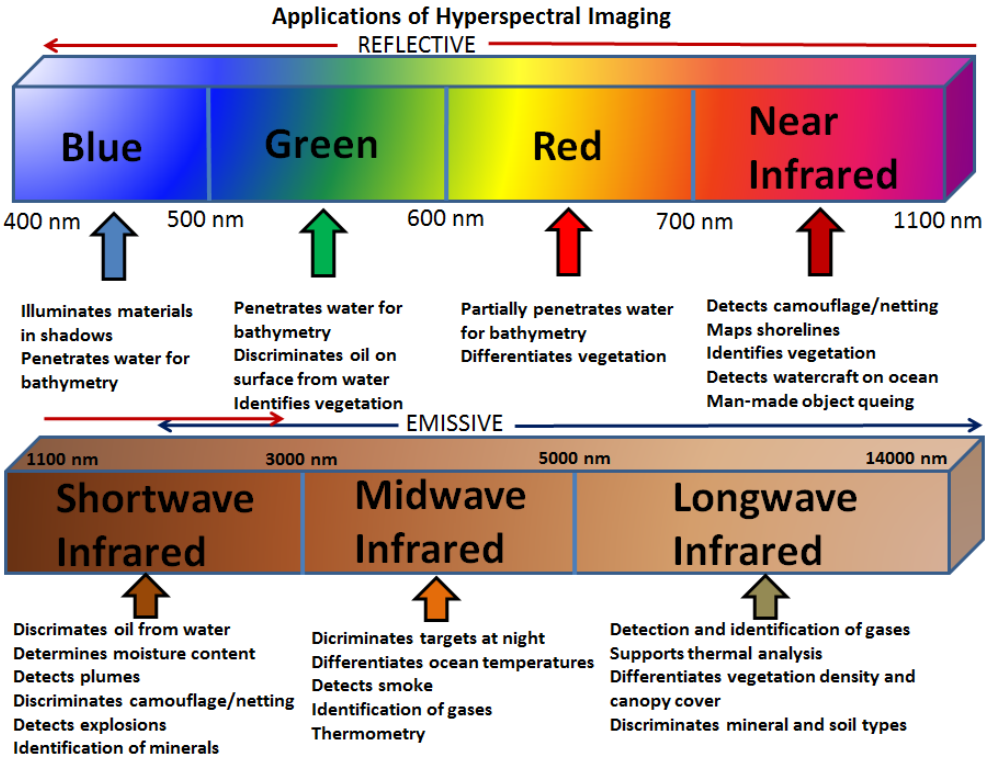


Fig. 2.3: Applications of Hyperspectral and Multispectral Imaging, [1].

Elowitz's diagram 2.3 depicts clearly the applications of images acquired at various ranges of the electromagnetic spectrum.

2.1.2 Hyperspectral Imaging Processing Techniques

We have emphasised that hyperspectral image data contains very fine spectral information. It also brings the challenge of redundancy with very high volumes of data. Thus efficient signal and image processing techniques have been necessary to be developed by academics and industry. The following is not an exhaustive list by any means, however it presents the most important techniques in image spectroscopy.

Compression

Hyperspectral image data cannot be compressed in the same manner as two dimensional images, i.e. spatial compression. The most important information lies in the spectral dimension, and thus more emphasis has to be put on spectral compression. The latest methodology applies sparse representation in the lossy hyperspectral image compression framework, improving classification performance [28].

Target Detection

Target detection refers to the techniques applied in surveilllance for identifying whether materials with known spectral signatures are present in an observed scene, [29]. This process consists of two stages: (i) identifying anomalies, i.e. detecting spectral vectors with significant spectral differences from their surrounding background pixels, and (ii) identifying whether the anomaly is in fact a target or just natural clutter, [30].

Classification

Feature mining is the important first step of hyperspectral image classification, i.e. separating the image into objects of interest. However some features of the original image may not be as effective at separating objects of interest, due to the high spectral redundancy in the data. Thus dimension reduction methods are necessary, [30]. Hyperspectral Image Segmentation is a supervised classification problem, which partitions the data into spatially-connected regions according to some predetermined criteria, such as spectral homogeneity, [31].

Resolution Enhancement

Resolution Enhancement, also known as Hyperspectral Data Fusion, combines the hyperspectral and multispectral images of the same scene. The aim is to fuse the high spectral resolution of hyperspectral images with the high spatial resolution of multispectral images, to achieve the best possible of both worlds, [30]. Super-resolution

techniques are related to pan-sharpening techniques in digital image processing.

Spectral Unmixing

Figure 2.2 portrays the ideal spectral unmixing problem where every pixel is assumed to be the "pure" spectral signature of a material. In reality, most pixels in a scene cannot account for the spectral signature of one material, due to the limited spatial resolution of hyperspectral images. Thus *mixed pixels* are the common occurrence in hyperspectral images, where each pixel's acquired spectral signature is a mixture of different materials with various contributions. Therefore spectral unmixing algorithms aim to identify and characterize the different materials mixed in a pixel, hence the term *unmixing* [5]. Spectral Unmixing is the main focus of this paper, and will be discussed in further detail in this paper.

2.1.3 Acquisition Modes and Scanners

Modern hyperspectral sensors are capable of sampling the electromagnetic spectrum into hundreds of spectral bands, i.e. specifically the visible, near-infrared, shortwave infrared, and mid-wave infrared spectral bands by measuring the radiance of the scattered electromagnetic energy in their field of view via hundreds of channels [32]. Hyperspectral imaging usually collects radiance in the range of wavelengths $0.4 - 2.5\mu m$. There are very few hyperspectral scanners that also collect data in the thermal infrared (TIR) range of $8 - 15\mu m$.

Figure 2.4 depicts the multispectral acquisition ranges. Even though multispectral scanners cover a wider spectral range (including TIR), they usually collect 3 to 15 very specific spectral bands. Each band is acquired using a remote sensing radiometer, which makes multispectral imaging very expensive.

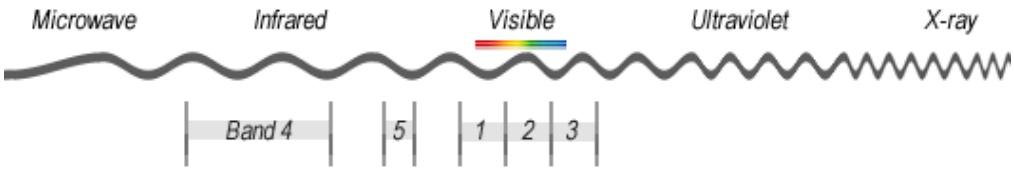


Fig. 2.4: Multispectral acquisition range, 5 bands example, not drawn to scale, source: GIS Geography

In figure 2.5 we see an example of the hyperspectral acquisition range, which is not drawn to scale. Hyperspectral images can have hundreds to thousands of bands,

because they are much narrower than multispectral bands, typically $10 - 20nm$.

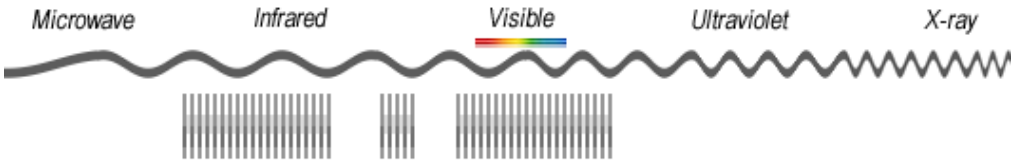


Fig. 2.5: Hyperspectral acquisition range example, not drawn to scale, source: GIS Geography

Table 2.1 lists the imaging spectrometers mounted on aircraft and spacecraft used to acquire hyperspectral image data. Here we can see that hyperspectral scanners usually acquire over 200 spectral bands in the $0.4 - 2.5\mu m$ wavelength range, i.e. much finer spectral resolution than the multispectral scanners part of USGS's Landsat program. Note that the altitudes in table 2.1 are average specifications, as the crafts do fly at different altitudes, which affect the spatial resolution. Also AVIRIS refers to "AVIRIS Classic", and not AVIRIS Next Generation.

Instrument	**Type**	**Org.**	**Alt. (km)**	**Spatial Reso. (m)**	**Spectral Reso. (nm)**	**Covers (μm)**	**Spec. Bands**
HYDICE	Airborne	NRL	1.6	0.75	7-14	0.4-2.5	210
AVIRIS	Airborne	NASA	20	20	10	0.4-2.5	224
HYPERION	Space	USGS	705	30	10	0.4-2.5	220
PRISMA	Space	ASI	614	5-30	10	0.4-2.5	238
CHRIS	Space	ESA	556	36	1.3-12	0.4-1.0	63
HyspIRI	Space	NASA	626	60	4-12	0.38-12	217

Tab. 2.1: Imaging Spectrometers, source: Jose Bioucas-Dias, GISTAM 2015 presentation

The Landsat 1-5 could only acquire 7 bands, and Landsat 7 collected 8 bands. The new Landsat 8 can acquire 11 bands, as shown in the following table:

Bands	**Wavelength** (μm)	**Resolution**(m)
Band 1 - Ultra Blue (coastal/aerosol)	0.43 - 0.45	30
Band 2 - Blue	0.45 - 0.51	30
Band 3 - Green	0.53 - 0.59	30
Band 4 - Red	0.64 - 0.67	30
Band 5 - Near Infrared (NIR)	0.85 - 0.88	30
Band 6 - Shortwave Infrared (SWIR) 1	1.57 - 1.65	30
Band 7 - Shortwave Infrared (SWIR) 2	2.11 - 2.29	30
Band 8 - Panchromatic	0.50 - 0.68	15
Band 9 - Cirrus	1.36 - 1.38	30
Band 10 - Thermal Infrared (TIR) 1	10.60 - 11.19	100
Band 11 - Thermal Infrared (TIR) 2	11.50 - 12.51	100

Tab. 2.2: Landsat-8 band specifications, source: Landsat-8

The TIR bands in table 2.2 are acquired at 100 meter resolution, but are resampled to 30 meter in delivered data product.

Some of the afore mentioned instruments have been decommissioned, for example the HYPERION in 2000. However their data is still available to download on the internet. More information can be found at the following web addresses:

- HYDICE: http://rsd-www.nrl.navy.mil/hydice;
- AVIRIS: http://aviris.jpl.nasa.gov;
- HYPERION (EO-1): http://eo1.usgs.gov;
- PRISMA: http://www.asi.it/en/flash en/observing/prisma;
- CHRIS: https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/;
- HyspIRI: http://hyspiri.jpl.nasa.gov;
- IASI: http://smsc.cnes.fr/IASI;
- Landsat-8: https://www.usgs.gov/land-resources/nli/landsat/landsat-8.

Table 2.3 illustrates the enormous sizes of *each* data cube acquired by these spectrometers, *assuming* each raw data point requires 16-bits, i.e. 2 bytes storage. These data quickly add up depending on the number of data cubes acquired in a scanning session, and thus makes storage and transmission a tremendous challenge.

Acquisition Instrument	**Size (samples x lines x bands)**	**Size (raw data points)**	**Size (Megabytes)**
HYDICE	200 × 320 × 210	13,440,000.00	25.63
AVIRIS	512 × 614 × 224	70,418,432.00	134.31
HYPERION	660 × 256 × 220	37,171,200.00	70.90
PRISMA	400 × 880 × 238	83,776,000.00	159.79
CHRIS	748 × 748 × 63	35,248,752.00	67.23
HyspIRI	620 × 512 × 217	68,884,480.00	131.39
IASI	765 × 120 × 8461	776,719,800.00	1,481.48

Tab. 2.3: Data Cube size calculations based on a data point requiring 16 bits storage in an ENVI file.

The number of samples, scanning lines and spectral bands in table 2.3 depend on the acquisition modes, of which there are two types mainly: *along-track* and *cross-track*.

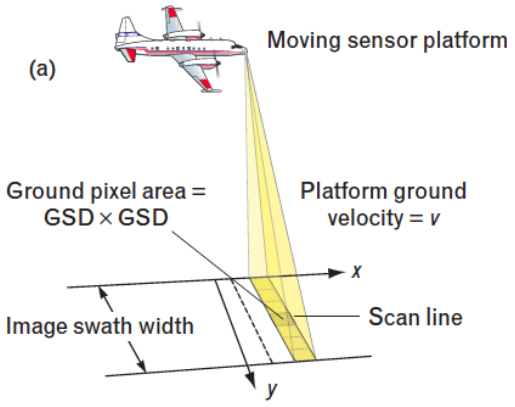


Fig. 2.6: Push broom scanner, source: Shaw 2003, [2]

Push broom scanners use the *along-track* acquisition mode, which works in in a similar fashion to photocopiers, document scanner and facsimile machines. The fixed line of sensors is arranged perpendicular to the flight direction of the aircraft or spacecraft used, see figure 2.6, which collect data from very high altitudes. These

scanners are the preferred acquisition mode and the specific campaign, [33]

Inside the hyperspectral camera there is a dispersing element used to "grate" the optical signal into the 200+ spectral bands corresponding to different wavelengths onto CCDs arrays, as explained in figure 2.7. These modern sensors can acquire hundreds of spectral bands.

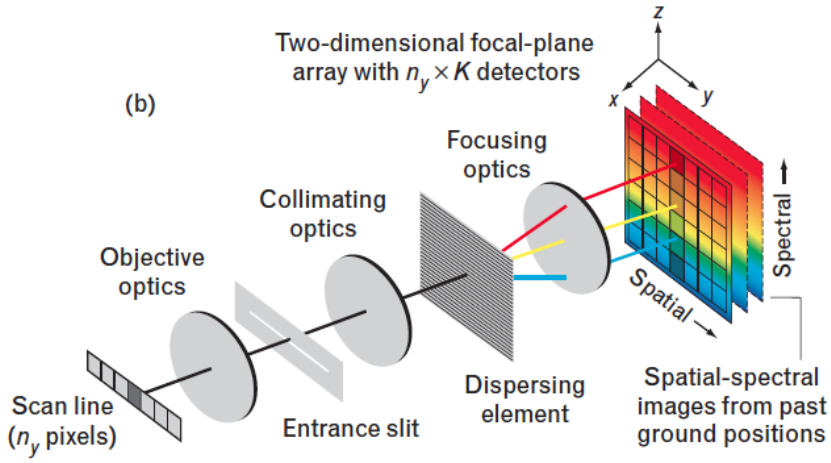


Fig. 2.7: Optical signal dispersion, source: Shaw 2003, [2]

Whisk broom scanners use the *cross-track* method, i.e. the sensor moves across the track along the swath width of the instrument, as the spacecraft flies forward. The detector can be stopped during the scan to focus on a particular subsection of the full swath, as shown in figure 2.8. This ability of the whisk broom allows for high spatial resolution on a particular subsection. Whisk broom scanners use bandpass filters to decompose the radiance in the corresponding spectral components, thus poses lower spectral resolution. This acquisition method is popular in multispectral imaging.

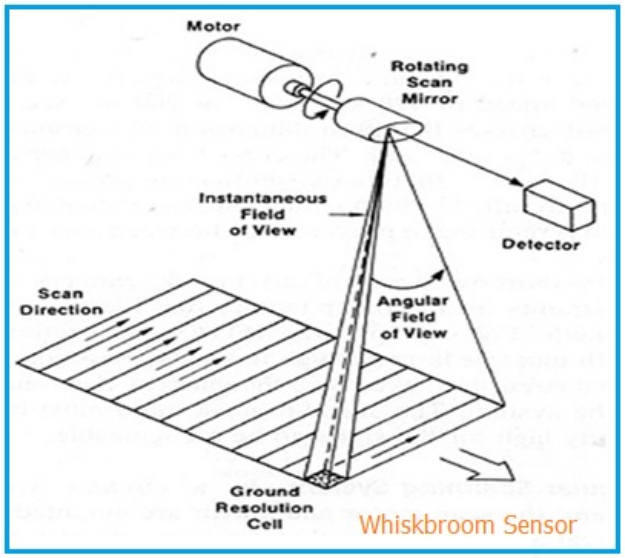


Fig. 2.8: Whisk broom scanner, source: RF Wireless

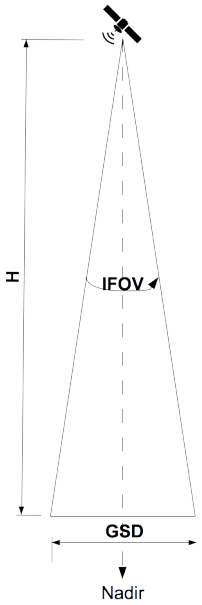


Fig. 2.9: IFOV and GSD

The collection system determines the size of the smallest recognised object on the ground. High spatial resolution allows for target detection, while high spectral resolution makes the identification and classification of materials possible, [33].

Spatial resolution depends on the operational altitude and the optics system. Scanners operating at very high altitudes, i.e. higher than 600 km above sea level, in general tend to have wider swath widths, with some exceptions, see 2.10. The high altitudes and wide swath widths collect cruder spatial information, resulting in larger pixel sizes with lower spatial resolutions due to wider *instantaneous field of view* (IFOV). The *ground sample distance* (GSD) gives us the distance between the centres of adjacent pixels, and thus an indication of the spatial resolution, see the calculations in table 2.4. Shaw and Burke [2] quantify the coverage tradeoff between spatial and spectral resolution in equation (2.1), as the *area coverage rate* (ACR) in terms of the GSD, i.e. pixel size on ground, the *signal-to-noise ratio* (SNR) associated with the error in the sensor, and the number of spectra channels B.

$$ACR \propto \frac{GSD^4}{SNR^2 \times B^2} \tag{2.1}$$

We can interpret the *spatial resolution* as inversely proportional to the ACR given in the relationship (2.1). Thus *spatial resolution* $\propto SNR^2 \times B^2/GSD^4$ indicates that the larger the pixel size on the ground, the poorer the spatial resolution.

Instrument	**Specifications**	$GSD = 2 \times H \times \tan(IFOV/2)$
LANDSAT7	Altitude H = 705 km, Swath width = 185 km, Spatial Resolution = **30** m, IFOV = 42.5 μrad, Spectral bands = 8.	$GSD = 2 \times (705 \times 1000) \times \tan((42.5/1000000)/2) = \mathbf{29.9630}\, m$
HYDICE	Altitude H = {1.6 km (operating), 6 km (designed for)}, Swath width = 0.25 km, Spatial Resolution = **0.75 - 3** m, IFOV = 507 μrad, Spectral bands = 210.	$GSD = 2 \times (1.5 \times 1000) \times \tan((507/1000000)/2) = \mathbf{0.7605}\, m$ $GSD = 2 \times (1.6 \times 1000) \times \tan((507/1000000)/2) = 0.8112\, m$ $GSD = 2 \times (6.0 \times 1000) \times \tan((507/1000000)/2) = \mathbf{3.0420}\, m$
AVIRIS	Altitude H = 20 km, Swath width = 11 km, Spatial Resolution = **20** m, IFOV = 1000 μrad, Spectral bands = 224.	$GSD = 2 \times (20 \times 1000) \times \tan((1000/1000000)/2) = \mathbf{20.0000}\, m$
HYPERION	Altitude H = 705 km, Swath width = 7.6 km, Spatial Resolution = **30** m, IFOV = 42.55 μrad, Spectral bands = 220.	$GSD = 2 \times (705 \times 1000) \times \tan((42.55/1000000)/2) = \mathbf{29.9980}\, m$
EnMAP	Altitude H = 653 km, Swath width = 30 km, Spatial Resolution = **30** m, IFOV = 45.00 μrad, Spectral bands = 220.	$GSD = 2 \times (653 \times 1000) \times \tan((45/1000000)/2) = \mathbf{29.3850}\, m$
HyspIRI	Altitude H = 626 km, Swath width = 120 km, Spatial Resolution = **60** m, IFOV = 96.00 μrad, Spectral bands = 217.	$GSD = 2 \times (626 \times 1000) \times \tan((96/1000000)/2) = \mathbf{60.0960}\, m$
Prisma	Altitude H = 614 km, Swath width = 30 km, Spatial Resolution = **30** m, IFOV = 48.34 μrad, Spectral bands = 238.	$GSD = 2 \times (614 \times 1000) \times \tan((48.34/1000000)/2) = \mathbf{29.6810}\, m$

Tab. 2.4: The sensor's operational altitude and IFOV are both built into the GSD.

In general, push broom scanners have poor spatial resolution, however whisk broom scanners have better spatial resolution because of the cross-track acquisition method. It seems to capture the spatial variability on the ground better.

For the optics system, the greater the number of CCDs in the sensor's two-dimensional arrays, the higher the spectral resolution. Push broom scanners posses two-dimensional arrays with several hundreds of CCDs along swath and the along-track dimensions. This allows push broom scanner longer exposure times on each scene due to its along-track sweeping, thus capturing higher amplitude radiometric signals. The whisk broom scanners have shorter residence times on a scene due to their cross-track sweeping, and hence are only able to capture weaker radiance energy. Also the whisk broom sensors have bandpass filters which limits the spectral wavelengths they acquire. The push broom optical system disperses the captured radiance on 200+ CCDs, thus higher number of spectral bands acquired. The wider the IFOV means the higher amplitude radiometric energy detected by the sensors. Greater number of CCDs on the sensor's arrays, longer exposure times and wider IFOV, in general imply higher spectral resolution. The issue with push brooms is that they scan different points on the ground at the same time, and due to the complex optics can cause spectral "smiles", [34], [35], [36].

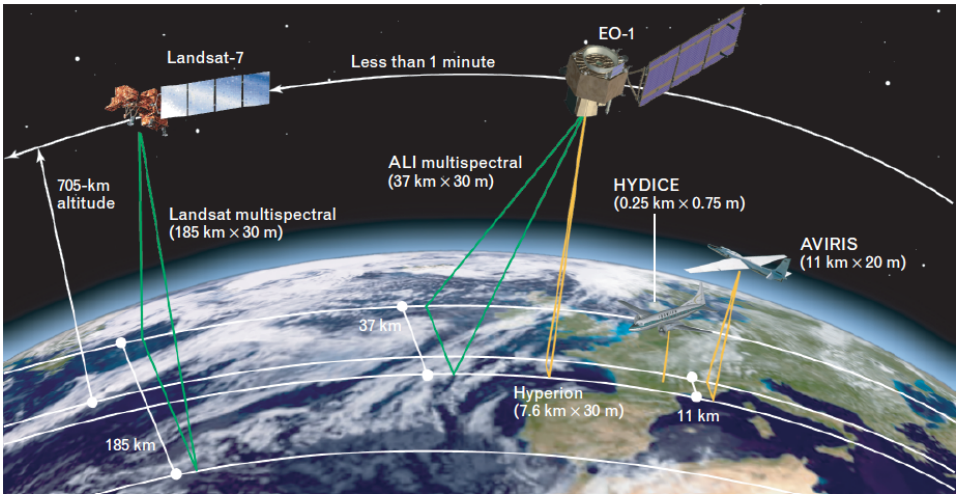


Fig. 2.10: Altitudes and area coverages of various scanners, source: Shaw 2003, [2]

2.1.4 Atmospheric Correction

All sensors are prone to inherent instrument errors and interfering noise, thus data should be spectrally and radiometrically calibrated first before analyzing it. NASA makes adjustments for geometric and radiometric errors in AVIRIS *raw data*, which are associated with the motion of the aircraft during the data collection. The calibrated radiance needs to be corrected for atmospheric effects and transformed to surface reflectance for further analysis [37].

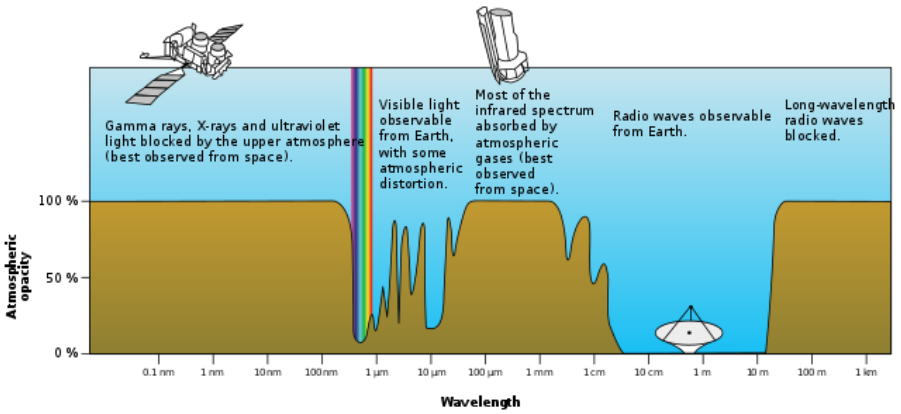


Fig. 2.11: Atmospheric electromagnetic transmittance and opacity, NASA.

The acquired raw pixel values are called *Digital Numbers* (DN) in hyperspectral unmixing jargon, commonly stored in ENVI files (acronym for "ENvironment for Visualizing Images"). These DN's are the uncalibrated 16-bit signed integers without physical units, that include the effects of surface reflectance, solar irradiance, atmospheric gas absorption, variation in illumination due to topography, and instrument response.

The *Radiance* is the amount of radiation scattered off the target area and off neighbouring areas. The calibrated radiance L_λ resulting from equation (2.2) [38] is free from detector sensitivity due to the Gain factor and the Offset, and has the physical units (Watts per (meter squared * micrometers)). Note the gain and offset are specific per spectral band. The gain and offsets are provided as metadata in ENVI files containing the data from AVIRIS.

$$L_\lambda = \text{Gain} \times DN + \text{Offset}. \tag{2.2}$$

The calibrated radiance still includes solar irradiance effects, which needs to be estimated next. The atmospheric effects at-sensor are complex. The remote optical sensor receives energy reflected from the Earth's surface transmitted through the *Top-Of-Atmosphere* (TOA), [3]. The TOA energy interacts with diffuse scattering, surface scattering, and scattering from the Earth's surface, as shown in figure 2.12.

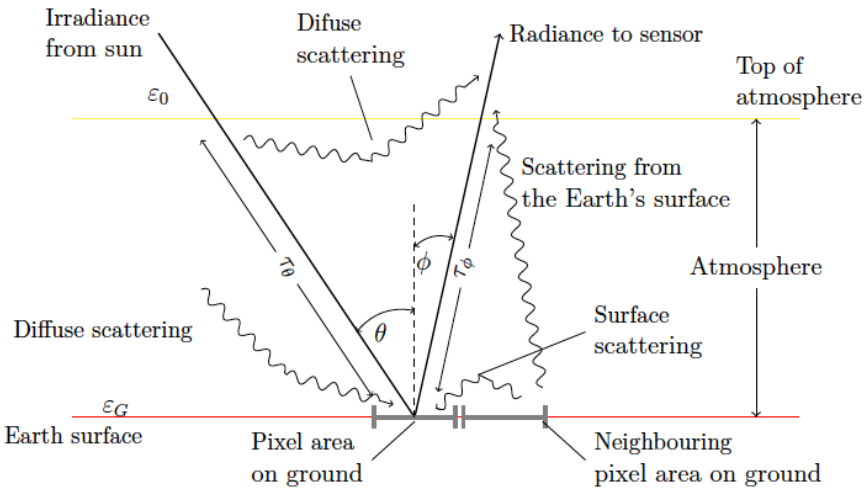


Fig. 2.12: At-sensor Top-of-Atmosphere reflectance interacting with scatterings, source: Bienarz [3].

Reflectance ρ_λ, i.e. TOA reflectance, is the proportion of radiation reflected off a surface to the incident solar radiation. Since hyperspectral sensors normally cover all the visible and near-infrared, that will include atmospheric absorption features and solar effects. Calibrating for these effects will emphasise the spectral features of the physical materials on the ground much more. The ENVI file needs to contain solar irradiance $ESUN_\lambda$ (Watts per (meter squared * micrometers)), sun elevation θ (degrees), and Earth-sun distance d (astronomical units), in addition to Gains and Offsets, to be able to transform Radiance L_λ to dimensionless Reflectence ρ_λ as in formula (2.3) [38]:

$$\rho_\lambda = \frac{\pi L_\lambda d^2}{ESUN_\lambda \sin\theta}. \tag{2.3}$$

In this paper we do not use ENVI files, instead we used pre-processed MATLAB data provided by Jose Bioucas-Dias. However this subsection informs the reader of this very necessary pre-processing step of calibrating the raw radiance data commonly used in scientific research, as we shall see later in order to identify endmembers from the observed data using ground-collected dictionaries, the data has to go through this calibration. Otherwise, the data collected on the ground obviously does not match data collected in space.

2.1.5 Spatial versus Spectral Resolution

Spatial resolution is a measure of the smallest discernible detail in an image,[39], meaning what is the smallest structure that we can recognise in the image. We have seen above, that in spectroscopy the wider the IFOV of the scanner and the higher the operational altitude of the scanner, the larger the pixel size or GSD, thus lower spatial resolution. The scanning method also matters here. The cross-track whisk brooms are better at capturing spatial variance.

Spectral resolution is defined as the ability to resolve features in the spectrum, i.e. important spectral characteristics of the signal. This is directly related to the number of spectral bands acquired, and can be understood in a similar fashion to audio signal and the number of harmonics that characterise that sound. The more harmonics, the "richer" the sound. Hyperspectral Imaging motivates the need for hundreds of contingent hyperspectral bands compared to the significantly smaller number of decoupled spectral bands acquired by Landsat 7, in figure 2.13. As we can clearly see in this figure, the Landsat scanner is missing a lot of critical spectral information.

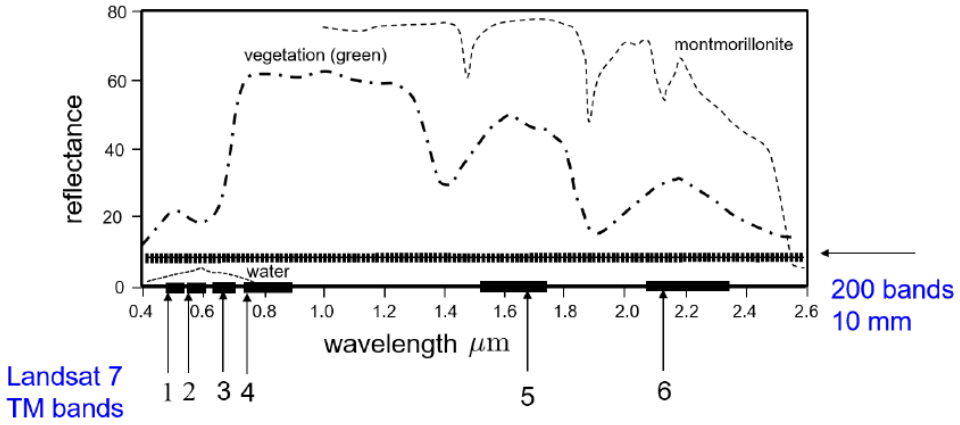


Fig. 2.13: Motivation for hyperspectral data, source: Bioucas-Dias, 2016 Grenoble presentation

Figure 2.14 demonstrates the high spatial resolution of multispectral images is suitable for land mapping, pattern recognition, and target detection. While figure 2.15 represents the hyperspectral data cube of the same scene as in 2.14, which

clearly has low spatial resolution and higher spectral resolution, which lends itself perfectly for material identification and classification, i.e. sparse unmixing. In the hyperspectral imaging techniques section 2.1.2, we mention *data fusion*. Yokoya depicts this concept very well in 2.16 of how resolution can be enhanced by taking the high spatial resolution of a multispectral image and the high spectral resolution of hyperspectral image of the same secen, and fusing them together.

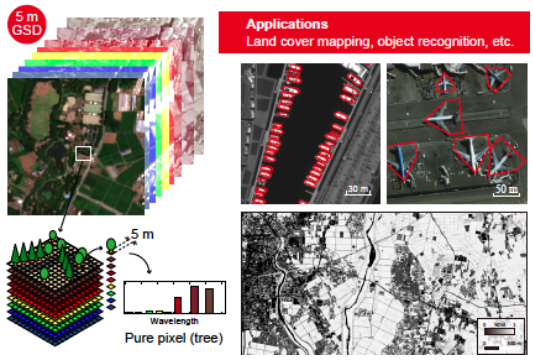


Fig. 2.14: Multispectral data cube, source: Yokoya [4]

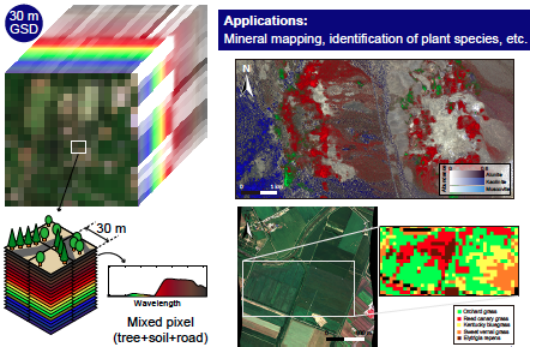


Fig. 2.15: Hyperspectral data cube, source: Yokoya [4]



Fig. 2.16: Fused data cube, source: Yokoya [4]

2.2 Mathematical Modelling

2.2.1 Basic Concepts

Vector Representation

The mathematical way of viewing "pure" pixels as in figure 2.2 is as hyperspectral vectors living in a lower-dimensional manifold, see figure 2.17. We shall see later on that this is the correct assumption, and thus the reason for reducing the dimensionality of the observed data.

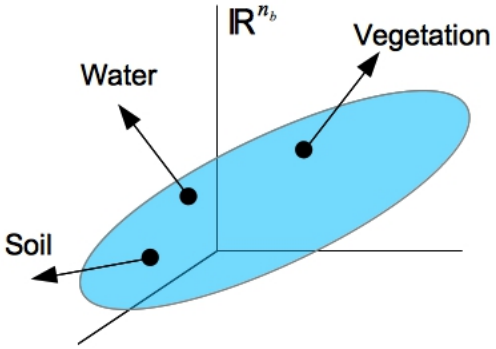


Fig. 2.17: Hyperspectral vectors in a lower-dimensional manifold, source: Bioucas-Dias, 2016 Grenoble presentation, modified

This vector representation allows for the use of linear algebra operations on the hyperspectral image data, for example norms.

Sparsity

The *support* of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is the index set of non-zero entries, [40]:

$$\text{supp}(\boldsymbol{x}) = \{j \in [1, n] : x_j \neq 0\} \tag{2.4}$$

Then we say that vector $\boldsymbol{x} \in \mathbb{R}^n$ is called k-*sparse* if at most k of its entries are non-zero, that is:

$$\|\boldsymbol{x}\|_0 = \texttt{card}(\text{supp}(\boldsymbol{x})) \leq k \tag{2.5}$$

where $\|\boldsymbol{x}\|_0$ is called the l_0-"norm" of vector $\boldsymbol{x}$, and the function $\texttt{card}(S)$ gives the *cardinality* of a set S, i.e. the number of its elements, [40].

Vector Norms

The vector norm l_p-norm, named after famous French mathematician Henri Lebesgue, for a vector $\boldsymbol{x} \in \mathbb{R}^n$, and $p \in \mathbb{R}$ is defined as:

$$\|\boldsymbol{x}\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i|^p} = (|x_1|^p + |x_2|^p + \ldots + |x_n|^p)^{1/p} \tag{2.6}$$

The most commonly used norm is the l_2-norm, also called the *Euclidean norm*, where $\|\boldsymbol{x}\|_2$ is the length of vector $\boldsymbol{x}$. The square of the l_2-norm can be written conveniently in terms of the dot product $\|\boldsymbol{x}\|_2^2 = \boldsymbol{x}^T\boldsymbol{x}$. The l_1-norm is called the *Manhattan norm*, where $\|\boldsymbol{x}\|_1$ gives the sum of the absolute values of all the entries in vector $\boldsymbol{x}$. The l_0-"norm" is actually a special case of l_p-norms since it is impossible to take the zero root, thus it is called the *pseudo* l_0-norm or the l_0-quasinorm, and it represents the number of non-zero elements of vector $\boldsymbol{x}$, defined as follows [40]:

$$\|\boldsymbol{x}\|_0 = \lim_{p\to 0} \|\boldsymbol{x}\|_p^p = \lim_{p\to 0} \sum_{j=1}^{n} |x_j|^p = \lim_{p\to 0} \sum_{j=1}^{n} 1_{\{x_j \neq 0\}} = \text{card}(\{j \in [1, n] : x_j \neq 0\}) \tag{2.7}$$

In the context of equation (2.7), if we have a zero element, we define $0^0 = 0$ such that we do not count zero elements.

The l_1-norm and the l_0-quasinorm have been made popular by the pioneers of *compressive sensing* such as Donoho, Candes, Tao, Romberg and Barniuk, [41], [42], [43], [44], [45], [46], [8], and these norms are central to sparse unmixing theory.

Matrix Norms

Considering a matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ the "entry-wise" matrix norms are defined similarly to the vector norms. The p-norm for $p \geq 1$ is defined as:

$$\|\boldsymbol{A}\|_p = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^p \right)^{1/p} \tag{2.8}$$

The more robust p, q-norms for $p, q \geq 1$ used in *sparse coding*, which will be discussed later on, are defined as follows:

$$\|\boldsymbol{A}\|_{p,q} = \left(\sum_{i=1}^{m} \left(\sum_{j=1}^{n} |a_{ij}|^p \right)^{q/p} \right)^{1/q} \tag{2.9}$$

When $p = q = 2$ the $L_{p,q}$-norm is called the **Frobenius norm**, which can also be defined as:

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\text{trace}(\boldsymbol{A}^T\boldsymbol{A})} \tag{2.10}$$

The Frobenius norm is basically the Euclidean norm for matrices, and it is easier to compute than induced norms, and is invariant under rotations. We shall see this norm used later on as well.

2.2.2 Assumptions and Conditions

Hyperspectral unmixing refers to any process that separates each pixel's frequency spectrum from a hyperspectral image into a collection of endmembers and a set of fractional abundances [5], i.e. one set per pixel.

The *endmembers* are generally assumed to be *pure* materials present in the scene, and the *fractional abundances* are assumed to be directly proportional to the percentage area covered by that particular material. Both concepts are subjective and problem dependent. For example, we might be interested in identifying the building structures, roads and vegetation present in a scene, where the side-by-side materials can be combined in a linear fashion with abundance weightings proportional to the area occupied. Or we might be interested in the chemical composition of the vegetation to identify the plant species found in the image, where the nonlinear combinations happen at microscopic level.

Mixed pixels are obtained due to insufficient *spatial resolution* of the imaging spectrometer, or due to *intimate mixing*. However, increasing the spatial resolution does not always fix the problem with intimate mixtures, as their combination of spectra occurs at a *microscopic* level. If increasing the spatial resolution helps us identify a few materials, then we say that the endmembers are mixed at *macroscopic* level.

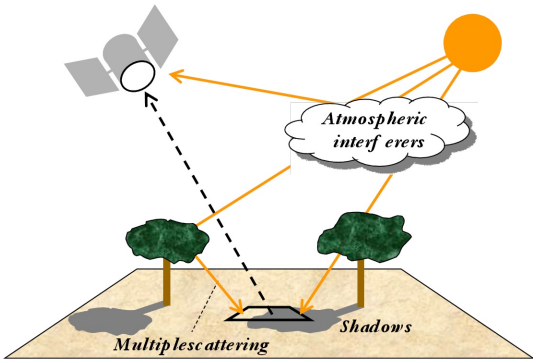


Fig. 2.18: Interference from tree tops, neighbouring obstacles, atmosphere, source: [5].

Even if the endmembers are pure, there are various interference types to consider, such as *atmospheric interference*, *multiple scattering* of light due to layers of obstacles on the ground, and *illumination interference* as some materials may be highly reflective and dominate the neighbouring materials, which induce errors into the measurements, see figure 2.18.

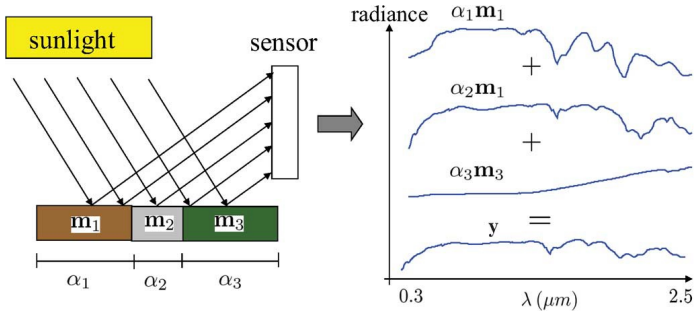


Fig. 2.19: Linear mixing illustration of measured radiance at a pixel. [5].

Thus we need to know what type of mixed pixels we are dealing with. Linear mixing only occurs at macroscopic level and the incident light interacts with just one material, such as in checkerboard type (*Lambertian reflectance* [47]) scenes in figure 2.19. The mixing occurs within the instrument itself, due to low spatial resolution.

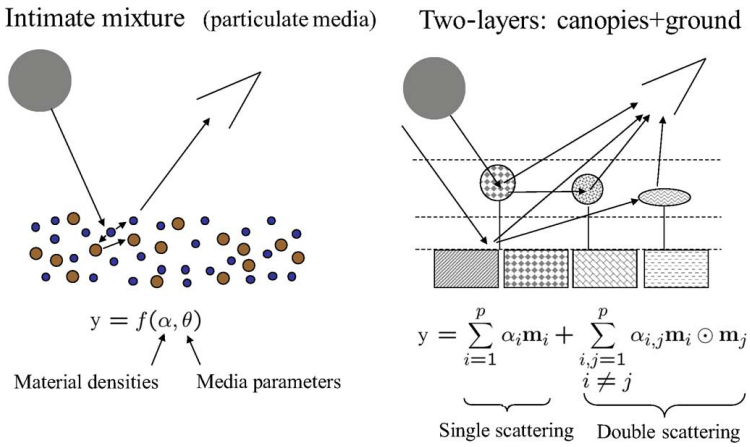


Fig. 2.20: Nonlinear mixing illustration of measured radiance at a pixel. [5].

In figure 2.19 the incidental light scattered by three materials is measured by a spectrometer which measures radiance in B spectral bands. The measured spectrum $\boldsymbol{y} \in \mathbb{R}^B$ at a pixel is the weight average of the endmembers $\boldsymbol{y} = \alpha_1\boldsymbol{m}_1+\alpha_2\boldsymbol{m}_2+\alpha_3\boldsymbol{m}_3$ where the alphas represent the fractional abundances.

At a microscopic level, we can obtain *intimate* homogeneous mixtures such as randomly and very closely compacted molecules in chemical composites, or *multilayered* mixtures such as light scattered by multiple layers of materials, as illustrated in figure 2.20. The nonlinear mixture in the image on the left-hand shows an intimate mixture of materials in close proximity, while the right-hand shows multiple interactions scattered off two layers. In general, nonlinear unmixing requires *a priori* knowledge about the geometry and physical properties of the observed objects [9].

Nonlinear Mixing Models are out of the scope of this report, however note there are numerous papers written on the subject such as [48], [49], [50], [51], [52], to name a few. This paper will assume the Linear Mixing Model (LMM), as it approximates real-life scenes reasonably well, and it is commonly the basis of many unmixing models including the sparse regression unmixing model.

The main conditions for the assumed linear mathematical model are as follows:

- The sum of the parts must add to the whole, i.e. the fractional abundances representing the proportions of endmembers in a pixels must add to one. This is not always imposed, but it is a fair condition.
- Abundance fractions cannot be negative. For example, we cannot have negative amounts of soil or vegetation in a scene.

2.2.3 Linear Mixing Model Derivation

Adopting the assumption of linear combinations of all the endmembers present in each pixel at the respective spectral band [9], implies that there is a linear relationship between the materials abundance coefficients and the surface area in the observed scene which they cover proportionally, and thus the reflecting surface can be portrayed as a "checkerboard" mixture where any package of incident radiation only interacts with one component [47].

Therefore if the surface of the observed scene is partitioned according to the fractional abundances as in figure 2.19, and the multiple scattering among distinct materials is negligible [5], then the spectrum of each pixel is well approximated by a linear mixture of endmembers, weighted by the corresponding abundances [47]. Therefore, each spectral measurement y_i at channel $i \in \{1, \ldots, B\}$ from a given pixel is given by the *linear mixture model* (LMM) in equation (2.11):

$$y_i = \sum_{j=1}^{p} m_{ij}\alpha_j + n_i. \tag{2.11}$$

where:

- y_i is the spectral quantity at a pixel for band $i \in \{1, \ldots, B\}$, regardless if the spectral quantity measures radiance or reflectance, the LMM mathematical formula still holds, although common practice makes use of reflectance;
- B is the total number of spectral channels according to the measurement instrument's technical specifications;
- p is the number of endmembers present in the scene, i.e. *pure* spectral signatures, where theoretically $p \leq B$, and real-life applications $p \ll B$;
- $m_{ij} \geq 0$ denotes the spectral measurement of endmember $j \in \{1, \ldots, p\}$ at i^{th} spectral band, i.e. the ground measured reflectance for j^{th} material responding in band i;
- α_j denotes the fractional abundance of jth endmember, i.e. the fractional area occupied by material j;
- n_i represents the additive perturbation such as noise affecting the measurement process and modeling errors.

In the more compact matrix notation, the LMM equation for a given pixel looks as follows:

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n} \tag{2.12}$$

where the terms in equation (2.12) have the following dimensions:

- Observed spectrum $\boldsymbol{y}$, i.e. the measured spectrum for each pixel, is a B-dimensional vector $\boldsymbol{y} \in \mathbb{R}^B$;
- Mixing matrix $\boldsymbol{M}$ is a $B \times p$ matrix which contains p column vectors as follows $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_p]$, where each column corresponds a material's spectral signature $\boldsymbol{m}_j = [m_{1j}, m_{2j}, \ldots, m_{Bj}]^T \in \mathbb{R}^B$;
- The $p \times 1$ fractional abundance vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_p]^T$ represents the weightings of the respective endmembers $\boldsymbol{m}_j$;
- Noise vector $\boldsymbol{n}$ is a $B \times 1$ vector accounting for measurement errors and such.

A quick note about the mixing matrix $\boldsymbol{M}$. In a typical hyperspectral unmixing scenario, we are given the set of observations $\boldsymbol{Y} = \{\boldsymbol{y}_d \in \mathbb{R}^B, d = 1, \ldots, N\}$, i.e. $\boldsymbol{Y} \in \mathbb{R}^{B \times N}$, where N is the number of pixels (i.e. we obtain $N = R \times C$ from the spatial dimensions of the image: number of rows R represents the *along-track* dimension of the moving sensor , and number of columns C represents the *swath* dimension of spaceborne sensor), and the goal is to estimate the mixing matrix $\boldsymbol{M}$ and the abundance vector $\boldsymbol{\alpha}$ for the LMM problem (2.12), [9].

Unmixing (2.12) is a *blind source separation* (BSS) problem, and BSS requires for the column vectors in matrix $\boldsymbol{M}$ to be *statistically independent*. This is however not the case for hyperspectral unmixing. The sum of the fractional abundances is constant, as per the condition above. Also the hyperspectral vectors are highly mutually coherent. Many materials will have similar spectral signatures. Instead of applying the traditional *independent component analysis* (ICA) to estimate mixing matrix $\boldsymbol{M}$, the common practice is to apply methods to extract the endmembers which are not based on statistical independence, but rather on convex geometry. We will see later in sparse regression we can use given libraries of spectral signatures, bypassing the endmember extraction process.

The fractional abundances are assumed to have a *linear* relationship with the areas occupied in a scene, they are subject to the following constraints:

$$\text{Abundance Nonnegativity Constraint:} \quad \alpha_j \geq 0,\ j = 1, \ldots, p; \tag{2.13}$$

$$\text{Abundance Sum-to-one Constraint:} \quad \sum_{j=1}^{p} \alpha_j = 1. \tag{2.14}$$

The "hard" *abundance nonnegativity constraint* (ANC) makes perfect sense practically, since we cannot subtract proportions of different materials, we can only add positive proportions. However numerically we might obtain negative coefficients, and thus the ANC is critical to impose on the problem.

The *abundance sum-to-one constraint* (ASC) is a "softer" condition to impose. The abundances don't necessarily have to add to one, since the algorithm might not be able to account for every material in a pixel.

The constraints in compact vector form, where $(.)^T$ denotes the transpose of a vector:

$$\text{Abundance Nonnegativity Constraint (ANC):} \quad \boldsymbol{\alpha} \geq \mathbf{0}; \tag{2.15}$$

$$\text{Abundance Sum-to-one Constraint (ASC):} \quad \mathbf{1}^T \boldsymbol{\alpha} = 1. \tag{2.16}$$

where the vector notation for constraints means that:

- the zeros vector $\mathbf{0}$ in ANC has $p \times 1$ dimensions;
- the ones vector $\mathbf{1}^T$ in ASC has $1 \times p$ dimensions.

The ANC and ASC constraints form the standard $(p-1)$-*simplex*, however it is more efficient to relax the sum-to-one constraint by applying "unconstrained" algorithms [9].

The main challanges for the LMM are: (i) to infer the materials matrix $\boldsymbol{M}$ from the observed data $\boldsymbol{Y}$ in some efficient way, and (ii) estimating the fractional abundances $\boldsymbol{\alpha}$ from the observed data and the inferred materials matrix, which is an *inverse problem* commonly solved by the *least squares* approach.

Let us now assume that the columns of $\boldsymbol{M}$ are *affinely independent*, meaning that $\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_p$ are all independent vertices of the $(p-1)$-simplex and thus the endmembers $\boldsymbol{m}_j$ do obey conditions (2.15) and (2.16), then referring to page 24 in Boyd's book [53], the set C:

$$C \equiv \left\{ \boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha} : \sum_{j=1}^{p} \alpha_j = 1,\ \alpha_j \geq 0,\ j = 1, \ldots, p \right\} \tag{2.17}$$

is the *convex hull* of the columns of $\boldsymbol{M}$, i.e. $C = \text{Conv}(\boldsymbol{M})$, which is the $(p-1)$-simplex in space $\mathbb{R}^B$.

In figure 2.21, Bioucas-Dias illustrates a 2-simplex $C = \text{Conv}\{M\}$ for a hypothetical mixing matrix $\boldsymbol{M}$ containing $p = 3$ endmembers. The green circles represent the spectral vectors, i.e. the measurements. The red circles represent vertices of the 2-simplex which correspond to the 3 endmembers in this example. This means that the green points are linear combinations of the red points, i.e. green points $y_i = \alpha_1 m_1 + \alpha_2 m_2 + \alpha_3 m_3 = [\alpha_1, \alpha_2, \alpha_3][m_1, m_2, m_3]^T = \boldsymbol{\alpha}_i \boldsymbol{m}_i^T$ in this example, with abundance coefficients $\alpha_j \geq 0$ for all j and $\sum_{j=1}^{3} \alpha_j = 1$. The axes represent the B spectral bands.

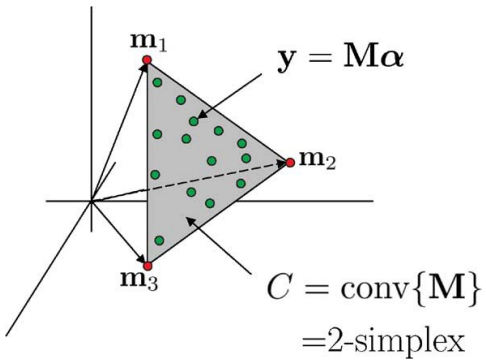


Fig. 2.21: Convex Hull C for $p = 3$ endmembers [5].

Note that the *geometrical* point of view of many endmember extraction algorithms is to infer the materials matrix $\boldsymbol{M}$, i.e. to identify the affinely independent **vertices** of simplex C, and it will be the basis of the algorithms presented in section 2.3.3. The convex geometry is also adopted in Bayesian statistical methods for extracting endmembers optimising estimators such as the *minimum mean squared error* (MMSE), *maximum a posteriori* (MAP) and *maximum likelihood estimator* (MLE), which are out of this paper's scope. The sparse unmixing approach bypasses the endmember extraction process by using spectral libraries available *a priori*, which is discussed in detail in section 2.4. The modern approach is to learn the dictionary from the observed data, called *sparse coding*, which is not the focus of this work.

The Linear Mixing Model given by equation (2.12) with conditions (2.15) and (2.16) formulate the problem for the remainder of this work.

2.3 Hyperspectral Unmixing Processing Chain

The basic idea of linear mixture modelling is that there is a small number of distinct materials in a given scene, that have relatively "constant" spectral properties. Although the endmembers' properties are not exactly the same for spectral signatures collected on the ground as those collected by airborne instruments. The goal of processing hyperspectral data is one and the same, to "unmix" the mixed pixels recorded by the spectrometer.

Bioucas-Dias et al. [5] illustrates in figure 2.22 that there are many routes that one can adopt in the processing of the remotely sensed data, and there are four major steps involved. The endmember extraction step and solving the inverse problem are the most challenging in this processing chain.

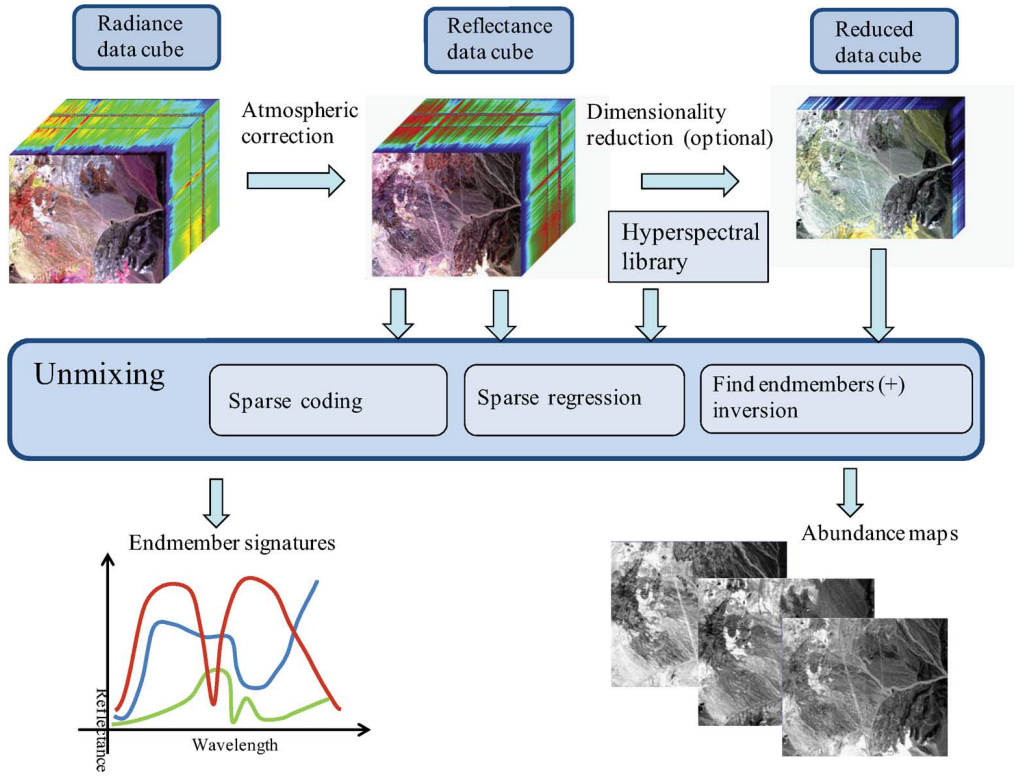


Fig. 2.22: Hyperspectral Unmixing Processing Chain [5].

Bioucas-Dias et al. provide the following descriptions of the four major steps depicted in diagram 2.22 :

1. **Atmospheric correction** compensates for atmospheric scattering effects of light by converting radiance into reflectance, where reflectance is dimensionless since it is reflectance by irradiance, see section 2.1.4 above. Note that linear unmixing can be carried out directly on radiance data, however it is customary to work with dimensionless calibrated reflectance data, as the spectral properties are much clearer.

2. **Data reduction** handles computations of high dimensional hyperspectral data more efficiently, and since there is great redundancy in the spectra, most information is preserved. Popular *Dimensionality Reduction* methods such as Principal Components Analysis (PCA), Orthogonal Subspace Projection (OSP), Minimum Noise Fraction (MNF) transformation, Hyperspectral Signal Subspace Identification by Minimum Error (HySime), etc. identify the appropriate subspace, thereby improving computational performance, and reducing complexity and data storage.

3. **Unmixing** is the key step of identifying the endmembers lying in the *subspace* and their corresponding fractional abundances *at each pixel*. If the subspace has p-dimensions (i.e. p endmembers identified or assumed), i.e. $\mathbb{R}^p$, then we usually work with a $(p-1)$-simplex, i.e. *convex hull*. This concept was presented in the mathematical modelling section 2.2.3. There are three general approaches to *Endmember Extraction*:

    - *Geometrical* approaches, which will be discussed in detail in this paper, attempt to optimise the unmixing problem by fitting a convex hull around the data which lies in a lower-dimensional subspace, and it is assumed that the *vertices* of the simplex *are* the endmembers, thus the name "endmember" for spectral signatures found at the "ends" of the convex hull. The most popular geometrical approaches are N-FINDR [54], Pixel Purity Index (PPI), and Vertex Component Analysis (VCA), upon which more efficient approaches have been built. The *abundance maps* are estimated in the "inversion" step.
    - *Statistical* approaches are applied to highly intimate mixtures, such as chemical, geological, etc., which can only be modeled as *nonlinear mixture models* (NMM). Geometrical methods yield poor results for highly mixed pixels. However, statistical methods represent variability in endmembers naturally. The statistical framework for spectral unmixing is formulated as a statistical inference problem, usually following the Bayesian

approach. These frameworks have high computational complexity, especially if we include neural networks into the framework. Statistical approaches are not discussed in this paper, since we only consider linear mixing models (LMM).

- *Sparse Regression* approaches [55], [9], [16], [5], which will be the main focus of this paper, formulate the unmixing problem as a *linear sparse regression* problem in a similar way to *compressive sensing* [41], [46], [8]. This framework relies on the existence of spectral libraries, also called *dictionaries*, usually acquired in the laboratory, all available from organisations such as NASA, USGS, and many more. Since dictionaries are potentially large, they need to be "pruned" to smaller subsets. Spectral libraries and data are acquired under different conditions, so radiometric calibration is required. Dictionaries can also be *learned* from the recorded data, termed as *sparse coding*. Dictionary pruning and sparse coding are advanced topics, which we only touch on in this paper.

4. **Inversion** is the last step of estimating the abundance maps using the *least squares* approach. The inversion step solves a constrained optimization problem given the observed spectral vectors and the identified endmembers, i.e. the inferred spectral signatures, by minimizing the residual between these two terms. The fractional abundances are often constrained to be *nonnegative* and must *sum to one*, i.e. they belong to the probability simplex. A popular method for computing the abundance maps is the *fully constrained least squares* (FCLS) method [56]. The benefit of *sparse regression* unmixing approach [9] which assumes the data is *sparse*, is that the endmember extraction and inversion steps are implemented simultaneously. The hyperspectral unmixing problem is NP-hard [57] due to its high dimensionality and the complexity of identifying pure spectral signatures, thus it is a combinatorial problem which will be discussed in more detail in section 2.4.

Next we discuss these four steps in much more detail. This literature review will describe the algorithms generally used for running the experiments in this work's research scope.

2.3.1 Dimensionality Reduction - Lower Subspace Projection

It is perfectly reasonable to assume that the hyperspectral signal lives in a *lower* dimensional space which is clouded by noise, i.e. the number of endmembers present in a given scene is often much smaller than the number of bands B. The *intrinsic* dimensionality is the minimum number of parameters needed to account for the

observed properties of the data [7]. There is a lot of redundancy in the spectral information. The true dimensionality of the data in a remotely sensed scene is difficult to determine due to restricted access to some exploration areas. Some hyperspectral data poses low spatial resolution such as $30 \times 30\,m^2$ per pixel, making it difficult to discriminate materials. However modern hyperspectral sensors can capture remote scenes with much better spatial resolution of $2 \times 2\,m^2$.

We can approximate the *virtual dimension* coined by Chang and Du [58], [59], i.e. the minimum number of spectral signatures, which may be larger than the intrinsic dimension, if we assume that the linear mixing model (LMM) is a good approximation. This means that the spectral vectors lie in or very close to the lower dimensional space.

Reducing the high spectral dimensionality is not a necessary step for unmixing of the hyperspectral data [47], however dimensionality reduction may improve computational time, and lower complexity, data storage size and signal-to-noise ratio (SNR) [5]. This is an extremely helpful pre-processing step.

To distinguish between *dimensionality reduction* and *compression*, the latter aims at reducing the dimension of data with the purpose of the efficient *reconstruction* of the original signal [47], where some degree of redundancy is acceptable. Dimensionality reduction seeks the best *representation* of the original signal with the *minimum amount of relevant information* in the lowest dimension without redundancy, see figure 2.23. The majority of the data lost should be noise, and the data retained must explain the *variance* in the data, i.e. the important information only.

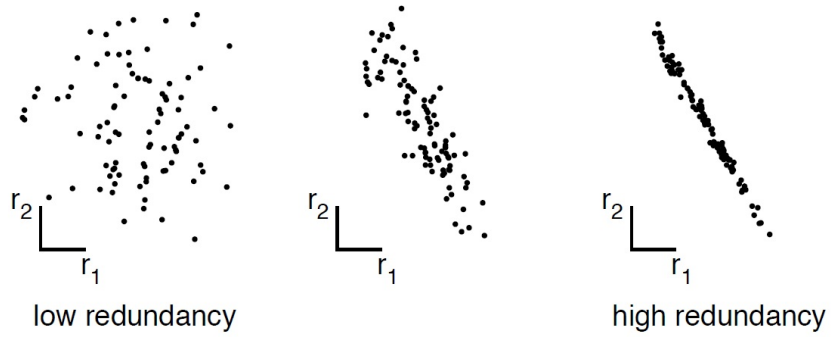


Fig. 2.23: Redudancy levels explained by Shelns [6] for two separate measures r_1 and r_2 .

Thus spectral unmixing utilizes *orthogonal projection* techniques to identify the best subspace to represent the data by optimizing an objective function, such as the general Lebesgue norm (L_p), defined in section 2.2.1.

Neighbouring hyperspectral vectors in real-life scenes have high *spatial correlated*, but it is the *spectral correlation* that allows the translation to lower subspaces, i.e. we can discriminate between materials with similar spectral signatures to each other. Thus orthogonal projection is required to transform the observed coordinate system to a new subspace which is an *independent* orthogonal coordinate system, which explains the data in fewer components doing away with redundancy and thereby reducing dimensionality.

We discuss next the popular projection methods, such as *principal component analysis* (PCA), *maximum noise fraction* (MNF), and mention a few others.

Principal Components Analysis

Principal component analysis (PCA) identifies the orthogonal axes, i.e. basis vectors to project the data onto without the loss of information, see figure 2.24. The new basis vectors are mutually uncorrelated. These axes represent the variance in the data in order of magnitude. Higher variance σ^2 represents the significant endmembers, while very low or close to zero variance represents noise.

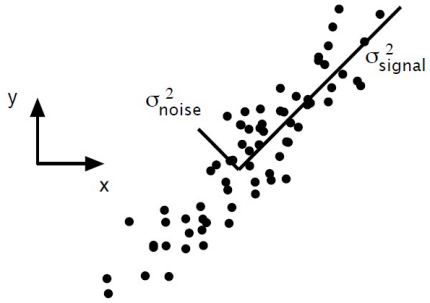


Fig. 2.24: PCA explanation given by Shelns [6].

PCA takes as input the observed spectrum for the high dimensional spectral vectors $\boldsymbol{y}_d$ for each pixel $d = 1, \ldots, N$ and performs an *eigendecomposition* of the *covariance matrix* (2.18) of $\boldsymbol{Y} \in \mathbb{R}^{B \times N}$, [47], [6]:

$$\hat{\Gamma}_{\boldsymbol{Y}} = \frac{1}{N} \sum_{d=1}^{N} (\boldsymbol{y}_d - \hat{\boldsymbol{\mu}}_{\boldsymbol{Y}}) (\boldsymbol{y}_d - \hat{\boldsymbol{\mu}}_{\boldsymbol{Y}})^T \tag{2.18}$$

where $\hat{\boldsymbol{\mu}}_{\boldsymbol{Y}}$ is the mean vector of the pixel set $\boldsymbol{Y}$

The resulting eigendecomposition can be expressed as follows:

$$\hat{\Gamma}_{\boldsymbol{Y}} = \boldsymbol{E}\boldsymbol{\Lambda}\boldsymbol{E}^T \tag{2.19}$$

where $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_p]$ is the $N \times p$ matrix of normalized eigenvectors, i.e. the orthogonal basis vectors, a.k.a. *principal components*, and $\boldsymbol{\Lambda}$ is the diagonal matrix of corresponding eigenvalues, i.e. variances, in order from the highest to the lowest value. To reduce the dimension $\boldsymbol{Y}$ we just need to premultiply each row, i.e. spectral signature, by $\boldsymbol{E}^T$ to move $\boldsymbol{y}$ to the new system of decorrelated variables oriented along the eigenvectors of $\boldsymbol{E}$ [47].

Thus p is the number of nonzero eigenvalues which gives us the efficient lower dimension. Keshava and Mustard [47] found that just a hand-full of cumulative normalized eigenvalues, i.e. $p \ll B$, are usually sufficient to account for the majority of the total energy in a scene. Thus the lower-dimension vector conveys the most energy in the original higher dimension observations.

At least in principle, the p resulting PCA components in $\boldsymbol{E}$ are ordered in descending order of data variance's magnitude, however it only extracts relevant information content, and discards the noise content.

Maximum Noise Fraction

While PCA is fairly good at retaining energy for the relevant components of the original signals, it does not produce eigenvectors that optimize signal-to-noise ratio [47]. The *maximum noise fraction* (MNF) algorithm, which also goes by the name *noise adjusted principal components* (NAPC), is a linear transformation that consists of two PCA rotations and a noise whitening step [60].

First apply noise whitening using the noise covariance matrix, to linearly transform the random noise into *white noise* with unit variance. Then apply the PCA to decorrelate and rescale the noise in the data. The resulting noise is uncorrelated and has unit variance, i.e. the new noise covariance matrix is the identity matrix. Next apply PCA a second time to the noise-whitened data. The MNF algorithm

maximises SNR rather than the information content.

The principal components identified by MNF are not necessarily orthogonal, but they are identified and ordered in terms of decreasing signal-to-noise ratio.

HySime

Hyperspectral signal identification by minimum error (HySime) finds the first k eigenvectors that contain most of the information data via a minimum *mean squared error* (MSE) between the original data and its projection onto the subspace. The HySime algorithm performs eigendecomposition unsupervised and fully-automatic, i.e. it does not require any tuning parameters. As shown by Bioucas-Dias in [61], HySime algorithm performs better than most subspace identification algorithms.

The method estimates the signal and the noise correlation matrices. It then selects a subset of eigenvectors that best represents the subspace in a minimum MSE sense. This in turn minimizes the signal projection power, reduces dimension of the subspace, and minimizes the noise projection. HySime handles noise more efficiently.

Other Dimensionality Reduction Methods

PCA and MNF are statistical component analysis methods for reducing high dimensionality. Another such method is independent component analysis (ICA), whose goal is to separate a multivariable signal into a linear sum of non-Gaussian components, assuming statistical independence [60]. This popular blind source separation tool unfortunately does not work well on hyperspectral data since the sum of the abundances is constant, it implies statistical dependence [5].

The *optical real-time adaptive spectral identification system* (ORASIS) is a non-statistical methods developed by the U.S. Naval Research Laboratory for dimensionality reduction and endmember extraction using a modified Gram-Schmidt process to identify orthogonal basis [5]. ORASIS is a series of several hyperspectral processing modules. Each new pixel collected in a scene is compared to an existing exemplar pixel, and if the angle distance between the pixels exceeds a predefined threshold, then the pixel is added to the collection of exemplars. The orthogonal basis are calculated periodically from the set of exemplars.

It is worth mentioning that nonlinear spectral mixtures, which are out of the scope of this paper, replace low dimensional subspace with low dimensional manifolds. These require topological analysis, such as curvilinear component analysis, curvilinear distance analysis, and manifold learning [5].

2.3.2 Affine Projecting onto the Subspace

Using one of the dimensionality reduction methods presented above, we identify the subspace S with orthonormal basis $B \times p$ matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_p]$ where the columns $\boldsymbol{e}_i \in \mathbb{R}^B$ for $i = 1, \ldots, p$, meaning that every inner product $\langle \boldsymbol{e}_i, \boldsymbol{e}_j \rangle = 0$ for $i \neq j$ and $i, j = 1, \ldots, p$ and $\|\boldsymbol{e_i}\| = 1$. By projecting spectral vector $\boldsymbol{y} \in \mathbb{R}^B$ for any pixel onto S with respect to basis $\boldsymbol{E}$, we can rewrite the LMM equation (2.12) as:

$$\boldsymbol{E}^T\boldsymbol{y} = \boldsymbol{E}^T\boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{E}^T\boldsymbol{n} \tag{2.20}$$

$$\boldsymbol{y}_S = \boldsymbol{M}_S\boldsymbol{\alpha} + \boldsymbol{n}_S \tag{2.21}$$

with the projected signal $\boldsymbol{y}_S \in \mathbb{R}^p$ and projected noise $\boldsymbol{n}_S \in \mathbb{R}^p$, while the projected materials matrix $\boldsymbol{M}_S$ is a $p \times p$ matrix in equation (2.21).

Note that from here on, for simplicity reasons, we define $\boldsymbol{y} \equiv \boldsymbol{y}_S = \boldsymbol{E}^T\boldsymbol{y}$, $\boldsymbol{M} \equiv \boldsymbol{M}_S = \boldsymbol{E}^T\boldsymbol{M}$, and $\boldsymbol{n} \equiv \boldsymbol{n}_S = \boldsymbol{E}^T\boldsymbol{n}$ to be the *lower* dimensional quantities. Thus equation (2.21) will look similar to (2.12), keeping in mind the reduced quantities.

Bioucas-Dias states in [5] that projecting the original LMM (2.12) onto subspace S yields large computational, storage and SNR gains. These advantages are demonstrated in an earlier paper [61]. First two gains are obviously due to dimension $p \ll B$. The SNR gain approximately (B/p) dB, meaning that the projected noise is attenuated by a factor of p/B. Although the projection of the data onto the signal subspace attenuates the noise significantly, we shall see in The Inverse Problem section below, that dimensionality reduction does not improve the conditioning for the hypespectral unmixing inverse problem. Additional *denosing* algorithms are effective in the presence of very noisy signals.

Note that we can reconstruct the image at the original volume from the projected vectors $\boldsymbol{y}$, $\boldsymbol{n} \in \mathbb{R}^p$ and $\boldsymbol{M} \in \mathbb{R}^{p \times p}$ in the *simplified* notation described earlier to the original signal and noise vectors and mixing matrix with the higher dimensions by pre-multiplying with basis matrix $\boldsymbol{E}$, i.e. product $\boldsymbol{EM}$. It will be easier and

computationally more efficient to work with the simplified model in the reduced dimensions:

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n} \tag{2.22}$$

Model (2.22) is a simplification in a *lower* dimensional subspace. It therefore does not model pixel-to-pixel signature variability. The *shapes* of the endmembers are fairly consistent in the lower dimensions, however their *amplitudes* do vary in *scale* from pixel-to-pixel [5]. Therefore in the absence of noise, we have a matrix of endmembers $\boldsymbol{M}\boldsymbol{s}_i = [s(i,1)\boldsymbol{m}_1, \ldots, s(i,p)\boldsymbol{m}_p]$ for each pixel number $i = 1, \ldots, N$, where $s(i,j)$ is a positive scaling factor. This indicates that the scaling factor varies from pixel to pixel, and thus the observed spectral vectors are no longer in a simplex defined by a fixed set of endmembers, but rather in a *set* of linear equations accounting for each pixel index i:

$$\left\{ \boldsymbol{y}_i | \boldsymbol{y}_i = \sum_{j=1}^{p} \alpha_j s(i,j)\boldsymbol{m}_j,\ i = 1, \ldots, N \right\} \tag{2.23}$$

Therefore the coefficients $\boldsymbol{s}_i^T\boldsymbol{\alpha}$ need not sum-to-one, but must still be nonnegative. To estimate these coefficients maps is nearly impossible. We can use PCA to identify an *affine* set, i.e. set of independent orthogonal basis, that best represents the observed data in a *least squares sense*, and then project the lower dimensional data onto the resulting principal components, however these estimates suffer from rotation problems as PCA modifies the direction of the spectral vectors.

If we apply perspective projection, i.e. the *dark point fixed transfrom* (DPFT) [62] by projecting from origin so that the projected data lies on a hyperplane $\boldsymbol{v}^T\boldsymbol{u} = 1$ for any vector $\boldsymbol{v} \in \mathbb{R}^p$, and $\boldsymbol{u}$ is chosen such that our projected data $\boldsymbol{y}^T\boldsymbol{u} > 0$, then we encounter scaling errors $\boldsymbol{y}/(\boldsymbol{y}^T\boldsymbol{u})$, as shown by Bioucas-Dias in [5]. One way to eliminate errors is to discard projected vectors with large angles created between the projected and the unprojected vectors, if they exceed a predefined threshold angle. This whole transformation is called *affine projection*.

2.3.3 Geometrical Unmixing Methods

This section describes a big part of this research report. While there are alternative ways of extracting endmembers, such as statistical methods and dictionary learning, the geometrical endmember extraction algorithms are still common practice in the industry. The aim is to use convex geometry to infer the endmembers participating

in the observed data.

There are two categories of geometrical approaches to unmix the endmembers: *pure pixel* (PP) based and *minimum volume* (MV) based. Pure pixel approaches belong to the MV class as well, however they are subjected to the *pure pixel constraint* which will be explained below. The concept of minimum volume simplex was introduced by Craig in 1994 [62].

We revisit the convex hull C in (2.17) of the spectral unmixing problem, and the example of a 2-simplex in figure 2.21. Bioucas-Dias illustrates the concept of minimum volume with examples of 2-simplexes, i.e. $p = 3$ endmembers, for three different data sets in figure 2.25, source: Jose Boucas-Dias.

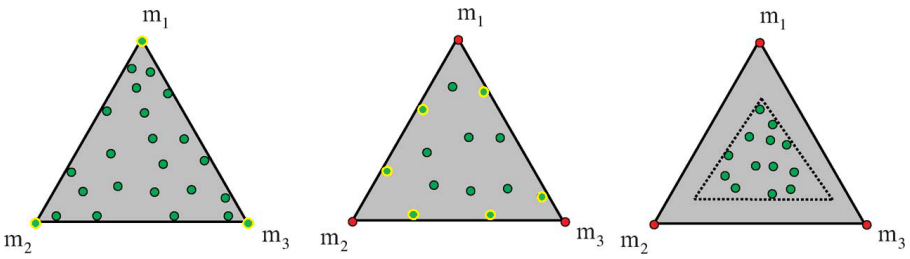


Fig. 2.25: Illustration of the concept of minimum volume concept [5].

Analyzing the simplex on the left in figure 2.25, notice that the three vertices correspond to the three endmembers $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$, i.e. for any p endmembers there exists at least one *pure* pixel that contains only the corresponding material. The simplex in the middle of figure 2.25 the data does not contain pure pixels but it contains at least $(p-1)$ spectral vectors on each facet of the convex hull.

In both of these cases the endmembers may be inferred by fitting a MV convex hull to the data. The left data set is the main idea of PP algorithms, while the middle data set is the basis for general MV algorithms. The MV simplex on the right of figure 2.25 corresponds to intimate nonlinear mixtures where none of the spectral vectors touch any of the facets of the simplex. These highly mixed data sets can only be unmixed within statistical frameworks.
The classical algorithms have been described in detail by Bioucas-Dias [5], Plaza [63], and Winter [64]. The following merely serves as a summary.

2.3.4 Pure Pixel Based Algorithms

The popular pure pixel (PP) based algorithms assume the presence of at least one pure pixel per endmember in the observed data, i.e. there is at least one spectral vector on each vertex of the data simplex. Thus however computationally efficient the *pure pixel constraint* may be, it is a hard constraint which may not hold for many data sets. Most PP algorithms find the set of most pure pixels in the data by wrapping a minimum volume convex hull around the data.

Pixel Purity Index (PPI) Algorithm

The widely used *pixel purity index* (PPI) algorithm uses MNF to reduce dimensionality and to improve SNR. PPI projects all the given spectral vectors onto *k-skewers*, see figure 2.26, where the number k is an input parameter, and it represents a large set of random vectors.

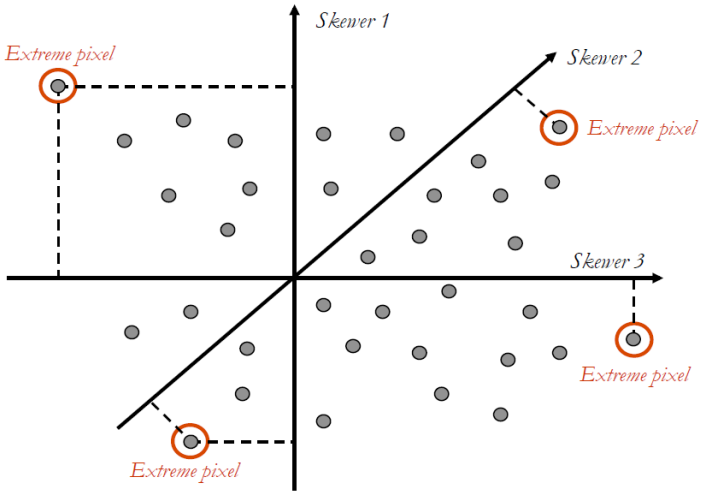


Fig. 2.26: PPI algorithm projecting data on k-skewers, source: gipsa-lab [7].

The algorithm stores the points which represent extremes for each skewer direction. Each pixel accumulates a score for the number of times an extreme is found. Those pixels that count above a certain *threshold*, another input parameter, are declared "pure" pixels.

The potential endmember spectra are loaded into a visualization tool displayed in the subspace identified by MNF, and rotated in real time until a desired number of endmembers q are visually identified as extreme pixels in the data cloud. Thus

PPI requires some manual intervention to select the final set of endmembers. Faster implementations with optimized parameter selection do exist [65] and more modern approaches of parallel computing [66], [67], [68] and [69].

N-FINDR Algorithm

N-FINDR developed by Winter [54] is one of the most popular endmember extraction algorithms in the remote sensing and geoscience industries. The volume defined by the simplex, i.e. convex hull, in the spectral dimensions is formed by the purest pixels in the vertices, and it is larger than the volumes defined by any other combinations of the observed pixels. N-FINDR thus inflates a convex hull inside the data, and finds the pixels defining the largest simplex volume, see figure 2.27.

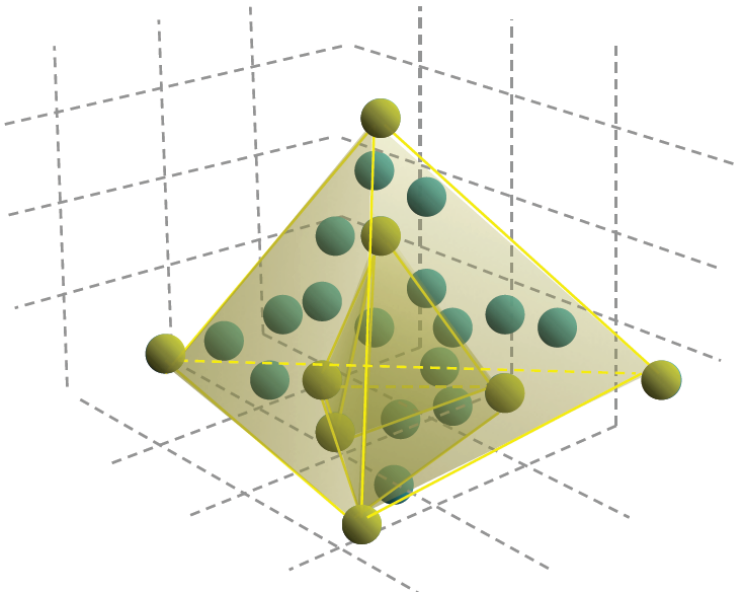


Fig. 2.27: N-FINDR algorithm inflates the volume of a convex hull inside data, source: gipsa-lab [7].

First, the MNF transform is applied to the observed data to reduce dimensionality. Next, N-FINDR randomly selects pixels to "qualify" as endmembers, i.e. pixels are initially evaluated as to their likelihood of being pure or nearly pure pixels, and a trial volume is calculated. We let $\boldsymbol{E}$ be the matrix of estimated endmembers, not to be confused with the orthogonal basis projection matrix $\boldsymbol{E}$ from earlier, defined as:

$$\boldsymbol{E} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \boldsymbol{e}_1 & \boldsymbol{e}_2 & \dots & \boldsymbol{e}_q \end{bmatrix} \tag{2.24}$$

where $\boldsymbol{e}_i$ are the endmember column vectors, and q is the number of endmembers used to calculate the convex hull volume, i.e. q is an input parameter.
The volume of the q-simplex $\boldsymbol{E}$ is proportional to its determinant, proved by Stein in [70], and thus Winter's formula for volume is given as:

$$V(\boldsymbol{E}) = \frac{1}{(q-1)!}\text{abs}\left(|\boldsymbol{E}|\right) \tag{2.25}$$

A trial volume is calculated for every pixel in each endmember position by replacing that endmember in matrix $\boldsymbol{E}$ and recalculating the volume. If the replacement results in a volume increase, i.e. "inflating" the convex hull, then that pixel replaces the endmember. This process is repeated until the algorithm goes through all the combinations are exhausted, and thus the largest convex hull volume is found.

In comparison to PPI which is partially interactive, Winter's N-FINDR algorithm is classified as fully-automated, and non-parameterized, i.e. does not require a threshold or some kind of stopping criteria. Performance improvements for the N-FINDR have been published in [71], [72], [73], [74], since exhausting all the possible combinations of pixels is computationally inefficient. However the advantage of N-FINDER being a fully-automated algorithm, is probably why it is more widely used than PPI.

2.3.5 Orthogonal Subspace Projection (OSP) Algorithm

The *orthogonal subspace projection* (OSP) algorithm which reduces dimensionality and extracts endmembers simultaneously, was developed by Harsanyi and Chang [75]. The OSP concept is similar to the material presented in the section 2.3.1. Harsanyi and Chang's algorithm's objective is to suppress signal interference in the least squares sense, and to maximize signal-to-noise ratio (SNR) for the spectral signatures of interest.
OSP first identifies the "brightest" pixel with the highest SNR and sets it as the first endmember. Then it applies an orthogonal subspace projection operator to all the pixels in the original image, to identify the most orthogonal pixel to the first one, i.e. setting it as the second endmember. OSP applies the operator again to all pixels to identify the most orthogonal pixel to the first two, thus setting it as the third endmember.

This process is repeated until all k endmembers are found, where k is an input parameter. In the process, the OSP also reduces dimensionality.

Vertex Component Analysis (VCA) Algorithm

Vertex component analysis (VCA) is based on subspace projections, assumes the presence of pure pixels, and it exploits the fact that endmembers are the vertices of a simplex. VCA iteratively projects data onto a direction orthogonal to the subspace already identified via PCA or MNF. The new projections correspond to the extremes of the projection, i.e. the affine projection of a simplex is also a simplex. This algorithm iterates until all endmembers are found. Nascimento shows in his paper [50] that VCA performs faster than PPI and N-FINDR.

Other Pure Pixel Based Algorithms

There are many other alternatives and improvements of OSP, PPI, VCA and N-FINDR, exploiting the geometrical properties of convex hulls, such as the *simplex growing algorithm* (SGA) which iteratively grows a simplex by looking for the vertices of the maximum volume, or highly iterative methods such as *iterative error anlaysis* (IEA). The IEA algorithm starts with the mean spectrum of the data as the initial endmember vector, then performs a series of linear constrained unmixings, choosing endmembers at each iteration that minimize the remaining error in the unmixed image. Not forgetting the U.S. Navy's ORASIS presented earlier which reduces dimensionality and extracts endmembers simultaneously.

The recent developments on geometrical endmember extraction algorithms are all linked to the classic pure pixel based algorithms above.

2.3.6 Minimum Volume Based Algorithms

MV method seeks the mixing matrix $\boldsymbol{M}$ which minimizes the volume of the $(p-1)$-simplex, i.e. the volume of $C = \text{Conv}(\boldsymbol{M})$ as in (2.17), applying soft or hard constraints. The pure pixel constraint is no longer en-

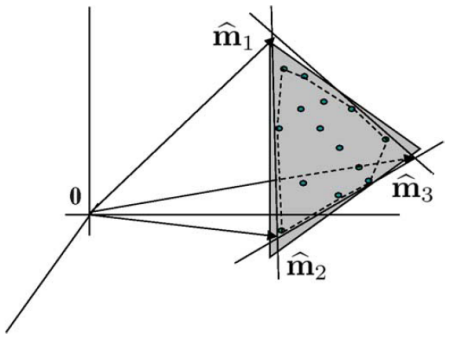


Fig. 2.28: Illustration of the minimum volume simplex concept [5].

forced, thus it is most likely the middle data set in figure 2.25. Figure 2.28 illustrates a similar data set to the middle one figure 2.25, but without enough points on the facets of the simplex, which represents a much harder nonconvex optimization problem [5]. MV concept is very important to statistical frameworks of extracting endmembers, and to understanding sparse regression unmixing methods which will be discussed ultimately.

We assume that our data has been projected onto a lower dimensional signal subspace S of dimension p, and that the material vectors $\boldsymbol{m}_i \in \mathbb{R}^p$, for $i = 1, \ldots, p$, are affinely independent, i.e. vectors $\boldsymbol{m}_i - \boldsymbol{m}_1$ for $i = 2, \ldots, p$ are linearly independent. Thus the simplex $C = \text{Conv}(\boldsymbol{M})$ has dimensionality $(p-1)$ such that volume of the simplex $V(\boldsymbol{M}) = 0$ in space $\mathbb{R}^p$ [5], since geometrical objects in a lower dimension have zero volume in a higher dimension. To obtain a nonzero volume in p-dimensions, the extended simplex $\hat{\boldsymbol{M}}_0 \equiv [\mathbf{0}, \hat{\boldsymbol{M}}]$ is considered. Thus the convex hull $C = \text{Conv}(\boldsymbol{M}_0)$ has the following volume $V(\hat{\boldsymbol{M}}_0) = \left|\det(\hat{\boldsymbol{M}})\right|/p!$ in p-dimensions according to Stein's formula [70].

The alternative volume computation consists of first shifting the data set to the origin, and working in the $(p-1)$-dimension, which then gives the following volume:

$$V(\boldsymbol{M}) = \frac{1}{(p-1)!}\left|\det\begin{bmatrix} 1 & \ldots & 1 \\ \boldsymbol{m}_1 & \ldots & \boldsymbol{m}_p \end{bmatrix}\right| \tag{2.26}$$

Craig's seminal MV algorithm [62] first identifies the signal subspace, and then it applies perspective projection DPFT to obtain shifted materials matrix $\hat{\boldsymbol{M}}_0 \equiv [\mathbf{0}, \hat{\boldsymbol{M}}]$, since the dark point fixed transform projects from origin $\mathbf{0}$. The algorithm iteratively changes each facet one at a time, while holding the other facets of the simplex fixed, such that the volume $V(\hat{\boldsymbol{M}}_0) = \left|\det(\hat{\boldsymbol{M}})\right|/p!$ is minimized, ensuring that all the spectral vectors belong to this simplex, [5].

In general the MV algorithms solve the following constrained and regularized optimization problem [7]:

$$\underset{\boldsymbol{M},\boldsymbol{S}}{\text{argmin}} \|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{S}\|_F^2 + \lambda V(\boldsymbol{M}) \tag{2.27}$$

$$\text{s.t.: } \boldsymbol{S} \succeq 0,\ \mathbf{1}_p^T \boldsymbol{S} = \mathbf{1}_N \tag{2.28}$$

where the data term $\|\boldsymbol{Y} - \boldsymbol{MS}\|_F^2$ minimizes the reconstruction error, with $\boldsymbol{Y}$, $\boldsymbol{S} \in \mathbb{R}^{p\times N}$, $\boldsymbol{S} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N]$, $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ and the volume term $\lambda V(\boldsymbol{M})$ promotes mixing matrices of minimum volume, whilst the Lagrangian multipliers λ control volume's relative weight, i.e. they are regularizes. We will arrive at this type of optimization problem shortly.

There are many versions of the MV algorithm each introducing more efficiency, such as the popular *minimum volume simplex analysis* (MVSA) [76] and *simplex identification via variable splitting and augmented Lagrangian* (SISAL) [77] algorithms. MVSA/SISAL is the *robust* implementation of the MV concept, since it penalizes *outliers* such as noise and spectral vectors lying outside the true data simplex [5].

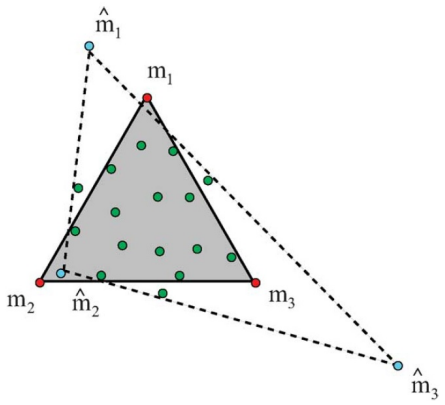


Fig. 2.29: Noisy data with outliers could lead MV to result in wrong estimate, i.e. dashed simplex [5].

Without penalizing noise and outliers, the MV algorithm presented above by Bioucas-Dias would probably result in the dashed estimate in figure 2.29. MVSA/SISAL first projects the data onto a subspace, and consequently projected materials matrix $\boldsymbol{M} \in \mathbb{R}^{p\times p}$ is square and theoretically invertible, such that:

$$\boldsymbol{M}^{-1}\boldsymbol{y} = \boldsymbol{M}^{-1}(\boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n}) = \boldsymbol{\alpha} + \boldsymbol{M}^{-1}\boldsymbol{n} \tag{2.29}$$

The MVSA part of the algorithm aims to minimize the volume:

$$V(\hat{\boldsymbol{M}}) = \left|\det(\hat{\boldsymbol{M}})\right|/(p-1)! \tag{2.30}$$

Minimizing the volume $V(\hat{\boldsymbol{M}})$ is equivalent to maximizing the volume of the

inverse of $\boldsymbol{M}$ [5]. Since $\det(\boldsymbol{M}^{-1}) = 1/\det(\boldsymbol{M})$ then we can change the minimization problem into a maximization problem and vice versa:

$$\begin{aligned}
\underset{\boldsymbol{M}^{-1}}{\operatorname{argmax}} \left|\det(\boldsymbol{M}^{-1})\right| &= \underset{\boldsymbol{M}}{\operatorname{argmax}}\ 1/|\det(\boldsymbol{M})| && (2.31)\\
&= \underset{\boldsymbol{M}}{\operatorname{argmax}}\ -\log|\det(\boldsymbol{M})| && (2.32)\\
&= -\underset{\boldsymbol{M}}{\operatorname{argmax}}\ \log|\det(\boldsymbol{M})| && (2.33)\\
&= \underset{\boldsymbol{M}}{\operatorname{argmin}}\ \log|\det(\boldsymbol{M})| && (2.34)\\
& && (2.35)
\end{aligned}$$

Li and Bioucas-Dias rewrite the MV problem in the simplified version, [76]:

$$\begin{aligned}
\hat{\boldsymbol{M}}^{-1} &= \underset{\boldsymbol{M}}{\operatorname{argmin}}\ \log|\det(\boldsymbol{M})| && (2.36)\\
\text{s.t.:}\quad & \boldsymbol{M}^{-1}\boldsymbol{Y} \succeq 0,\ \mathbf{1}_p^T\boldsymbol{M}^{-1}\boldsymbol{Y} = \mathbf{1}_N^T && (2.37)
\end{aligned}$$

Next they add the hinge parameter $\|\boldsymbol{X}\|_h \equiv \sum_{ij} h([\boldsymbol{X}]_{ij})$ where the *hinge function* is defined as $h(x) \equiv \max\{-x, 0\}$. Thus rewriting the problem in the final MVSA format:

$$\begin{aligned}
\hat{\boldsymbol{M}}^{-1} &= \underset{\boldsymbol{M}}{\operatorname{argmin}}\ \log|\det(\boldsymbol{M})| + \lambda\|\boldsymbol{M}^{-1}\boldsymbol{Y}\|_h && (2.38)\\
\text{s.t.:}\quad & \boldsymbol{M}^{-1}\boldsymbol{Y} \succeq 0,\ \mathbf{1}_p^T\boldsymbol{M}^{-1}\boldsymbol{Y} = \mathbf{1}_N^T. && (2.39)
\end{aligned}$$

Note that the new hinge term $\|\boldsymbol{M}^{-1}\boldsymbol{Y}\|_h$ in the MVSA equation (2.38) penalizes the negative components of $\boldsymbol{M}^{-1}\boldsymbol{Y}$ proportionally to their magnitude, since $\boldsymbol{M}^{-1}\boldsymbol{y} = \boldsymbol{\alpha} + \boldsymbol{M}^{-1}\boldsymbol{n}$ the hinge factor penalizes the negative α's, thus enforcing a soft constraint or *regularizer* [77]. The amount of regularization is controlled by the *regularization parameter* $\lambda > 0$. The soft constraints yield solutions that are robust to outliers and noise.

As the Lagrangian multipliers $\lambda \to \infty$, the soft constraint approaches a hard constraint [5]. The hard constraint paired with the fact that in practice, the inverse of $\boldsymbol{M}$ is hardly ever symmetric and positive-definite, and that the MVSA problem is most likely nonconvex, creates bigger problems which are optimized efficiently by solving a sequence of optimization problems using the method of variable splitting and augmented Lagrange multipliers, which is the SISAL part of the MVSA/SISAL

algorithm which is detailed in the paper [77].

The hard constraints $\lambda = \infty$ and the enclosing nonconvex simplex are solved by iterating through linear programs (LPs), which are part of the *minimum volume enclosing simplex* (MVES) algorithm. Even though the MVES problem deals with a nonconvex simplex, the paper [78] proves that the existence of pure pixels constraint is sufficient for this algorithm to identify the true endmembers. The more robust version of the minimum volume enclosing simplex (RMVES) [79] deals with the MVSA problem by taking into account noise effects in the observations by employing chance constraints, solving the MVSA by alternating optimization involving quadratic programming solvers. The chance constraints are probabilistic constraints that account for the randomness in the observed data, assuming that noise is Gaussian.

An alternative algorithm to the MV problem is the *minimum volume constrained-nonnegative matrix factorization* (MVC-NMF) [80], which overcomes the pure pixel constraint, and solves the following optimization problem without dimensionality reduction:

$$(\hat{\boldsymbol{M}}, \hat{\boldsymbol{S}}) = \underset{\boldsymbol{M},\boldsymbol{S}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{S}\|_F^2 + \lambda V^2(\boldsymbol{M}) \tag{2.40}$$

$$\text{s.t.:} \quad \boldsymbol{M} \succeq 0,\ \boldsymbol{S} \succeq 0,\ \mathbf{1}^T\boldsymbol{S} = \mathbf{1}_N^T \tag{2.41}$$

where the materials matrix is $\boldsymbol{M} = [\boldsymbol{m}_1, \ldots, \boldsymbol{m}_p] \in \mathbb{R}^{B\times p}$, and the fractional abundances matrix is $\boldsymbol{S} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_N] \in \mathbb{R}^{p\times N}$, and the norm $\|\boldsymbol{X}\|_F^2 = \operatorname{trace}(\boldsymbol{X}^T\boldsymbol{X})$ is the Frobenius norm of matrix $\boldsymbol{X}$, as defined in section 2.2.1, and λ is the regularization parameter.

MVC-NMF minimizes a two term objective function where the term $\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{S}\|_F^2$ measures the reconstruction error, and the squared volume term $V^2(\boldsymbol{M})$ is factorized in the paper [80], while the regularization term λ controls the tradeoff between the squared simplex volume and the reconstruction errors. The main difference is that MVC-NMF enforces the nonnegativity constraint, whilst MVSA/SISAL allows for the ANC to be violated.

While there are many more alternatives built on the minimum volume geometric concept, there is one more category worth mentioning, the *iterative constrained endmembers* (ICE) algorithm [81] which is similar to MVC-NMF, however the simplex volume term is replaced by the sum of squared distances between all the simplex

vertices, which transforms the nonconvex programming problem into a least squares problem that can be solved analytically.

Sparsity-promoting iterative constrained endmembers (SPICE) [82] is an extension of ICE, which incorporates sparsity-promoting priors to find the correct number of endmembers. The prior tends to discard the endmembers with zero abundance.

Both ICE and SPICE optimize an objective function of the following form:

$$f(\boldsymbol{M}, \boldsymbol{S}) = \frac{1-\mu}{N}\|\boldsymbol{Y} - \boldsymbol{M}\boldsymbol{S}\|_F^2 + \mu \operatorname{tr}(\boldsymbol{\Sigma}_{\boldsymbol{M}}) \tag{2.42}$$

where $\boldsymbol{\Sigma}_{\boldsymbol{M}}$ is the sample covariance matrix of the endmembers and $\mu \in [0, 1]$ is the regularization parameter. The *sparsity promoting term* (SPT) is usually incorporated into the Frobenius norm [82]. The solution is sensitive to the choice of μ as the best solution for $\mu = 1$ would be to find only one endmember, whilst $\mu = 0$ the best solution is one that yields zero errors.

Of course there are many more geometric based algorithms, such as *multiple endmember spectral mixture analysis* (MESMA) [83], *convex cone analysis* (CCA) [84], etc.. And there are also many Bayesian approaches to solving the MV problem, such as *piece-wise convex fuzzy clustering* [85] and *dependent component analys* (DECA) [86] algorithms, however these are beyond the scope of this paper.

2.3.7 The Inverse Problem - Estimating Abundance Maps

Our aim is to minimize the sum of squares for the LMM, i.e. the energy E or the l_2-norm squared of the *noise* in the acquired signal $\boldsymbol{y}$.

Referring back to the LMM equation (2.12), assume that we have estimated the number of endmembers, and that we have identified what those spectral signatures are, i.e. we now have the estimated materials matrix $\boldsymbol{M}$. From the observations $\boldsymbol{y}$ and the materials matrix, we obtain the noise residual or the reconstruction error $\boldsymbol{n} = \boldsymbol{y} - \boldsymbol{M}\boldsymbol{\alpha}$. This becomes a *least squares* (LS) problem whereby we minimize the energy of the noise or the error, seeking the best $\boldsymbol{\alpha}$ which minimizes this energy:

$$\hat{\alpha} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, E(\boldsymbol{\alpha}) = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, \|\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\alpha}\|_2^2 \tag{2.43}$$

This is the classic *inverse problem* in the spectral unmixing process whose solution can be approximated by the least squares method, but we have to assume that the matrix $\boldsymbol{M}$ has independent columns such that the matrix $\boldsymbol{M}^T\boldsymbol{M}$ is *invertible* and the approximate solution is *unique*.

Unconstrained Abundance Estimation

Unconstrained least squares (UCLS) approach is applied only if both the all sum-to-one constraint (ASC) and the all nonnegative constraint (ANC) are ignored. Thus traditionally the Moore-Penrose inverse $\boldsymbol{M}^+ = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T$ of the overdetermined materials matrix $\boldsymbol{M}$ is used to estimate which abundance coefficients vector is going to give us the minimum reconstruction error or noise:

$$\hat{\alpha} = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T\boldsymbol{y} \tag{2.44}$$

This is the least squares (LS) "solution". Note that this is really just an estimate, and most systems, such as MATLAB will use a Cholesky, QR or LU factorization to solve for the approximate solution, mldivide.

Nonnegative Constrained Abundance Estimation

To enforce the nonnegativity constrain on the least squares problem, since it would be unrealistic to have negative abundances, we rewrite the LS problem's objective function (2.43) for the LMM as a quadratic function including the ANC:

$$\hat{\alpha} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{\alpha} \tag{2.45}$$

$$\text{s.t.:} \quad 0 \leq \alpha_i \leq 1 \text{ for } i = 1, \ldots, p. \tag{2.46}$$

This *nonnegative least squares* (NCLS) problem is solved with quadratic programming solvers. The ASC is ignored as in a lot of spectral unmixing problems, the abundances do no add up to one exactly.

Fully Constrained Abundance Estimation

The *fully constrained least squares* (FCLS) approach rewrites the LS as quadratic programming problem as well, and both the ANC and the ASC are enforced, thus we have to solve the following quadratic problem:

$$\hat{\alpha} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{\alpha} \tag{2.47}$$

$$\text{s.t.:} \quad 0 \leq \alpha_i \leq 1 \text{ for } i = 1, \ldots, p; \tag{2.48}$$

$$\sum_{i=1}^{p} \alpha_i = 1. \tag{2.49}$$

The FCLS method of solving the inverse unmixing problem of evaluating the abundance map is the most popular method in the industry [56]. A lot of enhancements to the FCLS have been published [87], [88], [89], and many more, based on distance geometry.

The sum-to-one constraint can be easily satisfied by adding a row vector of ones to matrix $\boldsymbol{M}$ and an element one to column vector $\boldsymbol{y}$ [7], such as:

$$\tilde{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{M} \\ \mathbf{1} \end{bmatrix}, \quad \tilde{\boldsymbol{y}} = \begin{bmatrix} \boldsymbol{y} \\ 1 \end{bmatrix} \tag{2.50}$$

and thus FLCS can be treated as a NCLS problem.

Note that the "solutions" obtained for FCLS, NCLS or UCLS are called *abundance maps* since the abundance vector $\boldsymbol{\alpha}$ is estimated for every pixel.

Characterizing the Reconstruction Problem

Once we have reduced dimensionality of the original data, identified the endmembers of each pixel via a geometrical extraction method, and then estimated the corresponding abundance maps, then how do we know if we have good estimates?

To characterize the linear hyperspectral unmixing inverse problem, i.e. the reconstruction problem, we use the *signal-to-noise ratio* (SNR) [5]:

$$\text{SNR} \equiv \frac{\mathbb{E}[\|\boldsymbol{x}\|^2]}{\mathbb{E}[\|\boldsymbol{n}\|^2]} = \frac{\text{tr}(\boldsymbol{R_x})}{\text{tr}(\boldsymbol{R_n})} \tag{2.51}$$

where in paper [5], $\boldsymbol{x} \equiv \boldsymbol{M}\alpha$ represents the reconstructed spectrum for each pixel, $\mathbb{E}[\cdot]$ is the Expected value of the energy of the signal, and $\text{tr}(\cdot)$ is the trace of the reconstructed signal correlation matrix $\boldsymbol{R_x}$ and $\boldsymbol{R_n}$ [1].

Bioucas-Dias also introduces the *signal-to-noise ratio spectral distribution* (SNR-SD) measure which calculates the SNR along the direction of an eigenvector belonging to the signal correlation matrix $\boldsymbol{R_x}$:

[1] Correlation matrix $\boldsymbol{R}_X$ has elements $r_{ij} = \frac{1}{N}\sum_{n=1}^{N} X_{in}X_{jn} = \frac{1}{N}\{XX^T\}_{ij}$.

$$\text{SNR-SD}(i) = \frac{\lambda_{i,x}}{\boldsymbol{e}_{i,x}^T \boldsymbol{R_n} \boldsymbol{e}_{i,x}},\ i = 1, \ldots, p \tag{2.52}$$

where the pair $(\lambda_{i,x},\ \boldsymbol{e}_{i,x})$ represent the ith eigenvalue-eigenvector couple of matrix $\boldsymbol{R_x}$ for each reconstructed pixel x, ordered by decreasing value of $\lambda_{i,x}$. Therefore SNR-SD(i) yields the SNR along the direction $\boldsymbol{e}_{i,x}$. For this metric to represent an acceptable unmixing we require SNR-SD$(i) \gg 1$ for $i = 1, \ldots, p$.

2.4 Sparse Unmixing

Sparse unmixing solves the inverse problem, bypassing the endmember extraction step by using spectral dictionaries given *a priori*, and it estimates the sparsity of data by finding the *sparse* solution. First we visit basic concepts of norm regularization, whose importance will become evident for the discussion on sparse regression next.

2.4.1 Norm Regularization

The least squares approach in section 2.3.7 to solving a general linear system: $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}$, for measurements vector $\boldsymbol{y} \in \mathbb{R}^n$, matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and residual vector $\boldsymbol{\epsilon} \in \mathbb{R}^n$, will give the approximation as in 2.44:

$$\hat{\boldsymbol{x}}_{LS} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y} \tag{2.53}$$

The 2D geometrical interpretation of least squares in figure 2.30 shows that the estimate $\hat{\boldsymbol{y}}$ is a *projection* of measurement $\boldsymbol{y}$ onto the plane spanned by $\boldsymbol{A}$ with its columns $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$.

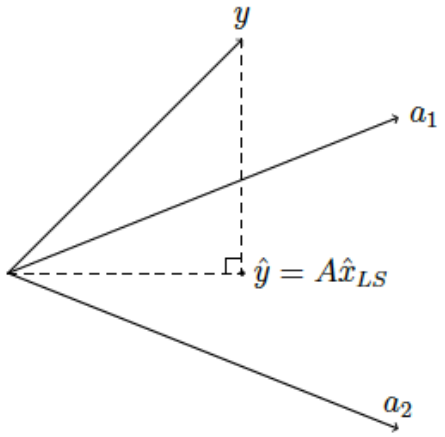


Fig. 2.30: Least Squares representation, source: Bieniarz [3]

l_2-Norm Regularization

The approximation in (2.53) is *ill-posed* because small amounts of noise lead to large errors in estimation, and *ordinary least squares* (OLS) tends to *overfit* the data with too many non-zero coefficients x_i, [3]. Thus we can introduce *ridge regression* where the least squares term $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2$ is *regularized* or *penalized* by the length of the estimated vector $\boldsymbol{x}$. Thus we rewrite the minimization problem as follows:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2 \tag{2.54}$$

where λ is the penalty parameter.

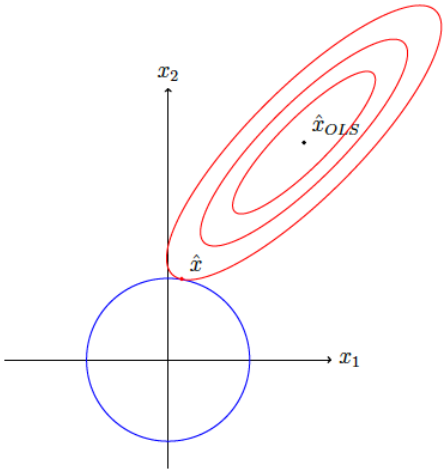


Fig. 2.31: Geometric interpretation of ridge regression, source: Bieniarz [3]

Figure 2.31 shows the geometrical interpretation of ridge regression in 2D. The blue circle represents the l_2-norm of the vector $\hat{\boldsymbol{x}}$, where $\hat{\boldsymbol{x}}_{OLS}$ is the solution to the ordinary least squares problem and $\hat{\boldsymbol{x}}$ is the solution to the ridge regression.

For a specific λ a solution with upper bound δ on the length of $\boldsymbol{x}$ is reformulated as:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_2 \leq \delta \tag{2.55}$$

where a small bound δ corresponds to a large penalty λ.

Thus we obtain $\hat{\boldsymbol{x}} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T + \lambda\boldsymbol{I})^{-1}\boldsymbol{y}$ as the approximation to (2.54) . If $\lambda = 0$ this approximation becomes the OLS approximation. Else if $\lambda \to \infty$ then solution shrinks to $\hat{\boldsymbol{x}} = \boldsymbol{0}$. Thus the linear model is balanced for $\lambda \in [0, \infty)$. The solutions are also restricted to the length of $\hat{\boldsymbol{x}}$.

However, due to the geometry of the l_2-ball, there is *no unique* solution $\hat{\boldsymbol{x}}$, as the ridge solution $\hat{\boldsymbol{x}}_{OLS}$ can land anywhere on the $\|\boldsymbol{x}\|_2$ ball.

l_0-Norm Regularization

The l_1-norm regularization does not promote sparsity. Thus if $\boldsymbol{x}$ is expected to be sparse, the l_0-norm defined in section 2.2.1, has to be used as a regularizer:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_0 \tag{2.56}$$

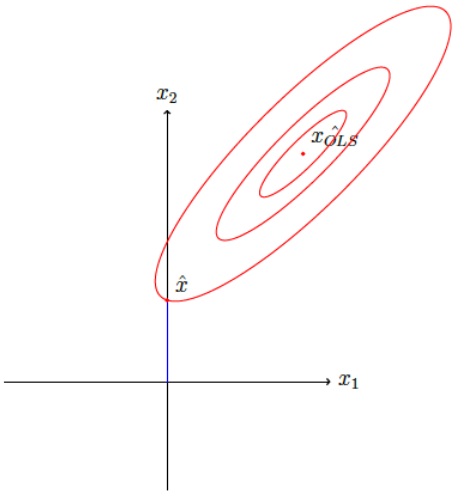


Fig. 2.32: Geometric interpretation of l_0-norm regularization, source: Bieniarz [3]

Minimization problem (2.56) is interpreted geometrically in figure 2.32. In theory the l_0-norm promotes sparsity, as the ridge of OLS problem should meet one of the $\hat{\boldsymbol{x}}$ "skewers". We can limit the number of non-zero elements in vector $\boldsymbol{x}$ by setting limit δ_0:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq \delta_0 \tag{2.57}$$

However if the sparsity of $\boldsymbol{x}$ is not known in advance, then we reformulate the problem by setting upper bound ϵ on the noise residual:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \epsilon \tag{2.58}$$

Unfortunately problem (2.58) is *non-convex* and it is usually very hard to solve such a combinatorial problem, [90], [8].

The 3D representation of the l_0 ball in figure 2.33 implies that the combinatorial search method would calculate the OLS approximation for every one of the 3 column vectors in matrix $\boldsymbol{A}$ combined with k non-zero entires in the solution. In higher dimensions for $\boldsymbol{A} \in \mathbb{R}^{n \times m}$, but there would be m coordinate axis or "skewers", and the OLS for $\boldsymbol{y} = \boldsymbol{A}_k \boldsymbol{x}_k$ would have to be calculated for all m skewers to choose k non-zero entries in $\boldsymbol{x}$, i.e. $\binom{m}{k}$ combinations.

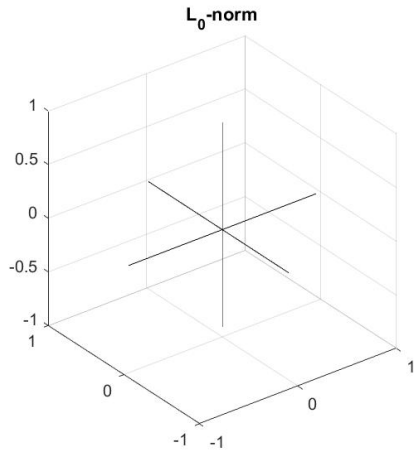


Fig. 2.33: Representation of the l_0-"ball" in 3D, which looks like skewers along the coordinate axis.

To illustrate how inefficient this problem is, say matrix $\boldsymbol{A}$ had $m = 400$ column vectors, and we guess sparsity $k = 5$, then the number of computations is $\binom{400}{5} \approx 8.32 \times 10^{10}$. Hard combinatorial problems are commonly solved by using *greedy search* algorithms, such as the *orthogonal matching pursuit* (OMP) algorithm which recovers k-sparse solutions. The OMP algorithm selects a column at a time from matrix $\boldsymbol{A}$ which is the most correlated to the current residual, and adds it to the set of active columns. Then OMP computes a new solution with the new set of active columns, i.e. hard thresholding. For more details refer to the chapter on algorithms 3. The computation time is cut down to only k iterations.

The issue with OMP is that it might select a wrong column due to columns of $\boldsymbol{A}$ being highly correlated, which is very common for hyperspectral signatures. OMP also doesn't perform well if the observations $\boldsymbol{y}$ are very noisy.

l_1-Norm Regularization

The alternative to the l_0-norm regularization for finding sparse approximations $\hat{\boldsymbol{x}}$ is to replace the regularization term with the l_1-norm, which is called *basis pursuit denoising* (BPDN), [91]:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1 \tag{2.59}$$

Substituting the l_0-norm penalisation term for the l_1-norm is that it transforms

a non-convex problem into a convex problem which is not as hard to solve. The geometrical interpretation is illustrated in figure 2.34.

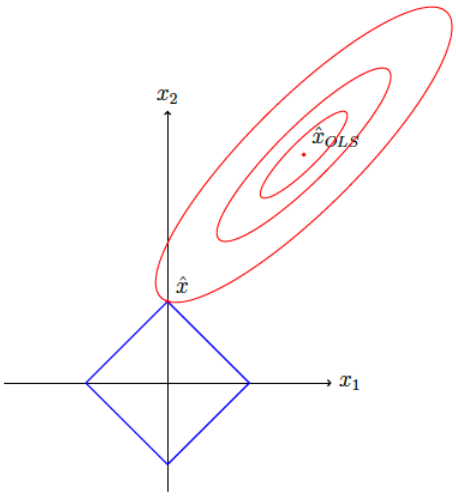


Fig. 2.34: Geometric interpretation of l_1-norm regularization, source: Bieniarz [3]

Candes [90] claims that a k-sparse solution is found by l_1-minimization, which can be proved to be *equivalent* to l_0-minimization under certain assumptions for matrix $\boldsymbol{A}$, such as the columns' mutual coherence should be small [92]. The l_1-norm is an octahedron in three dimensional space, and the sparse vector $\boldsymbol{x}$ lies on a coordinate axis, since there are far more zero coordinates. This vector is at one of the *vertices* of the octahedron. Thus a set of vectors $\boldsymbol{x}^*$ such that $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{y}$, the set of solutions projected onto $\boldsymbol{A}$, is a plane in three dimensions, which passes through a *unique* point $\boldsymbol{x}$, i.e through the vertex of the octahedron.

A plane $\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{y}$ that passes through a vertex is *virtually certain* to miss the interior of the cross polytope. This is what Candes calls the "miracle of high-dimensional geometry" that proves that the l_1-minimizer will "almost always" find the correct signal $\boldsymbol{x}$, [90]. Of course Candes, Romberg and Tao refer to measurement matrix as a randomly generated matrix in the context of *compressed sensing*, [90]. This is the same concept in sparse regression based hyperspectral unmixing. Baraniuk [8] illustrates the concept of l_1-minimization beautifully in figure 2.35.

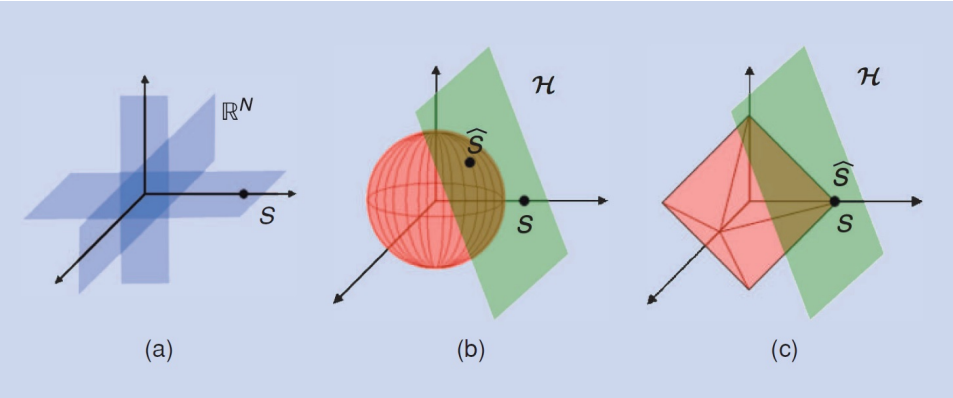


Fig. 2.35: (a) The subspaces containing sparse vectors in $\mathbb{R}^3$ lie close to the coordinate axes, which is a nonconvex problem. (b) The l_2-minimization fails to find sparse solution $\boldsymbol{s}$, as the point of contact $\hat{\boldsymbol{s}}$ projected on the green null-space can be anywhere on the red l_2-ball. (c) The l_1-minimizer finds the sparse point-of-contact $\hat{\boldsymbol{s}}$ to be the same as vertex $\boldsymbol{s}$ with high probability, due to the pointiness of the l_1-"ball". [8]

Consider the l_1-norm octaheadron in 3D, which guarantees that the solution will pass through one of the vertices. In higher dimensions the l_1-"ball" is called a cross polytope, which can best visualised as a sea urchin, and thus the solution will pass through a vertex of the l_1-"urchin".

The equation (2.59) can be expressed equivalently as the constrained version:

$$\underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_1 \leq \delta \tag{2.60}$$

where δ is the upper bound on the l_1-norm of the vector $\boldsymbol{x}$.

The BPDN algorithm works similarly to the OMP algorith, but instead of applying hard thresholding, it applies soft thresholding. That is, the algorithm updates vector $\hat{\boldsymbol{x}}$ with the largest possible step size until some other column of $\boldsymbol{A}$ has the most correlation with the current residual. More on this algorithm in chapter 3.

In summary, this section has provided the background on optimization methods that are commonly used in sparse regression, to be discussed in more detailed in direct application to the hyperspectral unmixing problem next.

2.4.2 Sparse Regression

The spectral unmixing methods, introduced circa 2009, approach the problem in a *semi-supervised* fashion, meaning that a collection of pure spectral signatures are known *a priori*. These libraries of spectral signatures are collected on the ground by field spectro-radiometers, where the collected data usually differs from the spectra collected by airborne instruments and satellites due to measurement conditions, thus requiring some calibration.

The assumption that the observed image signatures for each pixel can be expressed as linear combinations of a set of spectral signatures is at the heart of *sparse unmixing* (SU). The LMM mentioned previously constitutes the starting point for any sparse regression method:

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{n} \tag{2.61}$$

where as before dimensions for the spectra observed in each pixel are $\boldsymbol{y} \in \mathbb{R}^B$, and constituents are made up of the material matrix is $\boldsymbol{M} \in \mathbb{R}^{B \times p}$, the fractional abundances $\boldsymbol{\alpha} \in \mathbb{R}^p$, and the noise term $\boldsymbol{n} \in \mathbb{R}^B$, with B representing the number of spectral bands depending on the measuring instrument, and p the number of endmembers or materials. As mentioned previously equation (2.61) is assumed to apply for all N observed pixels.

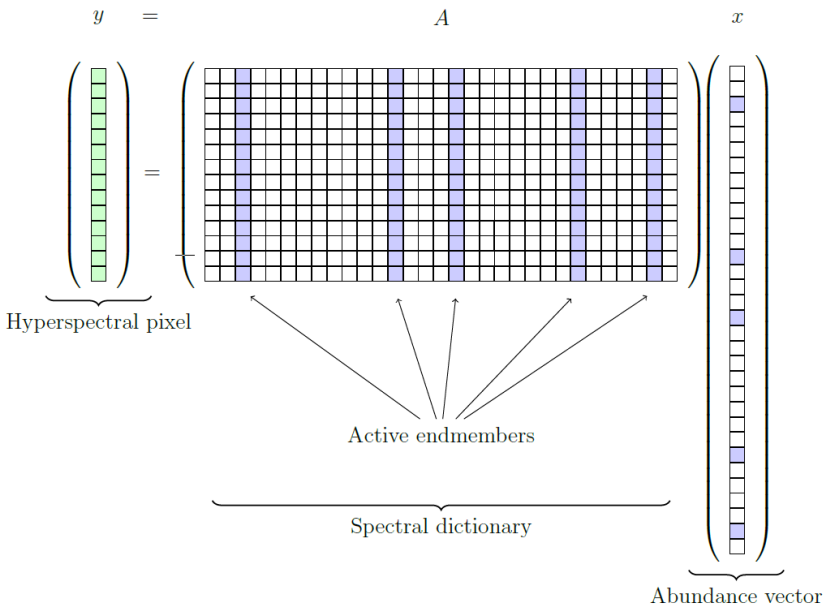


Fig. 2.36: Finding which columns in dictionary $\boldsymbol{A}$ are active in the image is a combinatorial problem, source: Bieniarz [3]

In constrast, the SU process uses available libraries, also known as *dictionaries* in SU jargon, which are known in advance. The availability of prior information thus makes it possible to circumvent the classic endmember extraction methods. Therefore we can rewrite the classic LMM model for the SU problem as follows, also illustrated in figure 2.36:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n} \tag{2.62}$$

where the dimensions are:

- the observed spectra for each pixel are $\boldsymbol{y} \in \mathbb{R}^B$ as before;
- the prior known dictionary of material spectral signatures collected on ground is $\boldsymbol{A} \in \mathbb{R}^{B \times m}$;
- the fractional abundances related to library $\boldsymbol{A}$ are denoted as $\boldsymbol{x} \in \mathbb{R}^m$;
- the noise term $\boldsymbol{n} \in \mathbb{R}^B$;
- B representing the number of spectral bands depending as above;
- and m is now the number of spectral signatures in library $\boldsymbol{A}$. In general $m \gg B$.

- Equation (2.62) applies for all N observed pixels.

The SU process aims at finding the optimal subset of signatures available *a priori* in a library $\boldsymbol{A}$ that can best model each pixel in a scene. In practice however, firstly the library can be potentially large, i.e. $m \gg B$, and the SU problem combinatorial problem which is NP hard, requiring efficient linear *sparse regression* (SR) techniques based on sparsity-inducing regularizers, similar to the description of the SPICE algorithm above.

The *sparsity* concept follows the same principles as *compressed sensing* [46], [44], [90], [8], which states that sparse signals are composed of a very small number of participating endmembers compared with the growing dimensionality and availability of spectral libraries [47]. Thus mixed pixels can be "compressed" down to linear combinations of just a few endmembers, call this very small amount k. Linear sparse regression algorithms are strongly linked to compressed sensing algorithms, and the direct application of compressive sensing to the spectral unmixing problem is an active area of research [93], [94], [95], [96].

2.4.3 Advantages of Sparse Regression over Classical Approaches

The practical issues with classical *endmember extraction* algorithms mentioned in the earlier sections of this paper, such as PPI, N-FINDR, OSP, VCA, are as follows according to Iordache [9]:

- Due to the low spatial resolution of the sensor, the classic algorithms may not be able to separate the different pure signatures well enough at macroscopic level, which may result in composites of individual pure spectra that occupy a pixel jointly. Thus the fractional abundances will not be correct since the endmembers may not be completely pure.
- Intimately mixed pixels can also appear as distinct materials, even though they are highly mixed at a microscopic level, independent of the spatial resolution of the sensor. Extracting the materials purely based on the image cannot accurately characterize intimate mixtures.
- The more advanced minimum volume techniques mentioned earlier such as MVSA/SISAL, SPICE, CCA, etc., have the necessary condition for $p - 1$ spectral vectors to be present on each facet of the simplex. This condition is very likely to fail for highly mixed pixels, and thus generating *artificial*

endmembers without physically meaningful spectral signatures associated to true materials.

- Sparse regression methods are more efficient in that they reduce the whole unmixing process to finding the optimal subset of spectral signatures in an available library. Thus the tedious process of extracting endmembers and its requirement of pure pixels in the data is no longer required.

2.4.4 Drawbacks of Sparse Unmixing

Naturally there are drawbacks to the attractive sparse unmixing methodology, and the following are the potential issues according to Iordache [9]:

- The spectral signatures in a library are rarely acquired under the same conditions as the airborne or satellite data. Conversely, image endmembers extracted via classic methods have the advantage of being at the same scale as the image data. Image endmembers may not be present in the library, and thus atmospheric correction is necessary. Sparse unmixing methods assume that the input data has been already converted from radiance to reflectance, and thus the atmospheric effect is removed.

- To obtain sparse solutions for undetermined systems it requires spectral signatures with low coherence, i.e. low correlation [41], [45]. Unfortunately in practice, spectral signatures are highly correlated. Fortunately, the number of materials participating in a hyperspectral scene is usually less than 20, and even more importantly, the number of materials participating in mixed pixels usually amounts to maximum 4 or 5 endmembers [47]. Thus the high sparse nature of hyperspectral signals mitigates the high correlation of hyperspectral libraries.

- The undetermined systems need to be solved as optimization problems with non-smooth terms [97] which cannot make use of gradient or Newton type of methods. These complex optimization problems need to be solved via fast alternating direction algorithms based on the *augmented Lagrangian method of multipliers* [98].

2.4.5 Sparse Regression Reformulation of the Unmixing Problem

Iordache [9] and Bioucas-Dias [5] reformulate the classic problem of unmixing the linear spectral model (2.62) as a semisupervised *sparse regression* (SR) optimization

problem. We assume that the available spectral library contains a potentially large number of spectral signatures *a priori*, which we no longer have to extract or generate from the original hyperspectral data via geometrical algorithms.

Problem (2.62) becomes a searching problem for the *optimal subset* of endmembers in a given library $\boldsymbol{A} \in \mathbb{R}^{B\times m}$ that best describe the original signal, where the library matrix $\boldsymbol{A}$ is undetermined due to $B < m$. Ideally, the search for the optimal number of nonzero components is performed using the l_0-norm since by definition [2] it finds the number of nonzero components in a signal, i.e. the sparsity of the signal.

Firstly, using the least squares approach, we can attempt to solve problem (2.62) by minimizing the reconstruction error:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad \text{s.t.: } \boldsymbol{x} \succeq \boldsymbol{0}, \tag{2.63}$$

where for each pixel in the original image with spectrum $\boldsymbol{y} \in \mathbb{R}^B$, the vector $\boldsymbol{x} \in \mathbb{R}^m$ represents the fractional abundances related to library $\boldsymbol{A} \in \mathbb{R}^{B\times m}$, which should be nonnegative to make physical sense. The underdetermined system (2.63) does not have a unique solution, and the solutions $\boldsymbol{x}$ are not sparse due to the "round" shape of the l_2-ball.

To take sparsity in consideration we need to regularize this problem by introducing Lagrange multipliers[98] to penalize the number of nonzero elements of $\boldsymbol{x}$, thus we get the regularized form of problem (2.63) :

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_0 \quad \text{s.t.: } \boldsymbol{x} \succeq \boldsymbol{0}, \tag{2.64}$$

where the fractional abundance vector $\boldsymbol{x} \in \mathbb{R}^m$ is assumed to be k-sparse, i.e. it has at most k nonzero components, and in theory, the l_0-norm should return a number no larger than k.

But we do not know in advance how sparse vector $\boldsymbol{x}$ is. Thus we can alternatively make the l_0-norm as the condition that the reconstruction error is subjected to:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \quad \text{s.t.: } \|\boldsymbol{x}\|_0 \leq \epsilon_0,\ \boldsymbol{x} \succeq \boldsymbol{0}, \tag{2.65}$$

However, problem (2.65) limits the number of nonzero elements. How do we know that we won't be missing out on nonzero components if the threshold ϵ_0 is set

[2] The L_0-norm is also known as the "nonzero counting norm" in compressed sensing jargon, see Donoho's definition in [44].

too small? Also the geometry of the l_2-ball is a sphere in three dimensions, and a disk in two dimensions. Finding the minimum Euclidean distance does not line up with finding sparsity.

Thus Iordache and Bioucas-Dias reformulate the SR problem with the l_0-norm as the objective function, since this is what we are seeking in the first place, i.e. the sparsity of $\boldsymbol{x}$:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_0 \quad \text{s.t.: } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \delta,\ \boldsymbol{x} \succeq \boldsymbol{0}, \tag{2.66}$$

The error tolerance parameter $\delta > 0$ may be due to noise and modeling errors. The solution of the $(m-1)$-probability simplex satisfies the error tolerance inequality $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \delta$.

Note that even though problem (2.66) is what we would ideally like to solve, this presents a combinatorial problem with a nonconvex simplex l_0-"ball" which is NP-hard to solve [57]. Next we present various ways to tackle this problem according to Iordache in [9]. The aim is to reformulate the SR unmixing problem into an "easier" problem to solve.

2.4.6 Exact Solutions in the Absence of Noise

In the absence of noise vector $\boldsymbol{n}$ we look for *unique* solutions to the following problem:

$$(P_0): \quad \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_0 \quad \text{s.t.: } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{2.67}$$

If the columns of matrix $\boldsymbol{A}$ are *independent and identically distributed* (i.i.d.) then $\operatorname{spark}(\boldsymbol{A}) = m + 1$ will guarantee a sparse unique solution $\boldsymbol{x}$ with the number of nonzero elements $k \leq B/2$, where the *spark* of a matrix is defined as the smallest k such that there exists a set of k linearly dependent columns of $\boldsymbol{A}$. Note that $\operatorname{spark}(\boldsymbol{A}) \leq \operatorname{rank}(\boldsymbol{A}) + 1$, [99]. Thus the spark could be used to obtain an estimate for the sparsity k of vector $\boldsymbol{x}$, however the spark is difficult to calculate for large matrices.

An alternative would be to get a lower bound, $\operatorname{spark}(\boldsymbol{A}) \geq 1 + 1/\mu(\boldsymbol{A})$, where $\mu(\boldsymbol{A})$ is the mutual coherence [3] of $\boldsymbol{A}$, i.e. the spectral angle difference between the

[3] Iordache's definition [9] of mutual coherence: $\mu(\boldsymbol{A}) \equiv \max_{1 \leq k,j \leq m, k \neq j} \frac{|\boldsymbol{a}_k^T \boldsymbol{a}_j|}{\|\boldsymbol{a}_k\|_2 \|\boldsymbol{a}_j\|_2}$.

columns of $\boldsymbol{A}$. This measure also proves useless since the mutual coherence for hyperspectral libraries is very close to one [9]. Iordache suggests that we could use the cumulative energy of the *discrete cosine transform* (DCT) coefficients of the library as an estimate for the spark's upper bound. Rather than attempting to estimate the spark of library $\boldsymbol{A}$, we adopt relaxed strategies for computing the solution for problem (P_0). Iordache illustrates two strategies: *pursuit algorithms* and *nonnegative signals*.

Since problem (P_0) is NP hard [57], we need to use *greedy algorithms* such as *orthogonal matching pursuit* (OMP) and *basis pursuit* (BP) which look for minimum of the l_0-"balls" along the orthogonal basis vectors, i.e. it has to pick an optimal axis in the illustration, which is very difficult.

Therefore the BP algorithm replaces the l_0-norm in problem (P_0) with the l_1-norm as follows:

$$(P_1): \quad \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_1 \quad \text{s.t.: } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}. \tag{2.68}$$

Problem (P_1) is solvable as it can be rewritten as a *linear programming* (LP) problem, and solved using LP solvers, because the l_1-"ball" is a *convex* simplex. The convex shape of the l_1-ball guarantees the solution will be through one of its vertices, [90] .

The basis pursuit (BP) algorithm to solve problem (P_1), under the tightest constraints α_k and $\beta_k > 0$ in the inequalities $\alpha_k\|\boldsymbol{x}\|_2 \leq \|\boldsymbol{A}\boldsymbol{x}\|_2 \leq \beta_k\|\boldsymbol{x}\|_2$, $\|\boldsymbol{x}\|_0 \leq k$, and $\gamma_{2s} \equiv \beta_{2s}^2/\alpha_{2s}^2 \geq 1$ ensures a unique s-sparse solution assuming $\gamma_{2s} < 4\sqrt{2} - 3 \approx 2.6569$, [9]. Iordache states the BP algorithm solves (P_1) faster and more efficiently than the OMP algorithm solving (P_0).

Restating problem (P_0) as a *nonnegative signals* problem, we will see that this enforces the ASC constraint indirectly:

$$\left(P_0^+\right): \quad \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_0 \quad \text{s.t.: } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y}, \quad \boldsymbol{x} \geq 0. \tag{2.69}$$

Iordache assumes that matrix $\boldsymbol{A}$ doesn't contain any zero vectors as its columns, then it is always possible to find a vector $\boldsymbol{h}$ ensuring the nonnegative constraint on the spectral library:

$$\boldsymbol{h}^T\boldsymbol{A} = \boldsymbol{w}^T > 0, \tag{2.70}$$

where all the components of vector $\boldsymbol{w}$ are nonnegative. We define diagonal matrix $\boldsymbol{W} \equiv \text{diag}(\boldsymbol{w})$ as a well defined matrix with positive diagonal entries, so that $\boldsymbol{W}^{-1}$ exists. Next we define $\boldsymbol{z} \equiv \boldsymbol{W}\boldsymbol{x}$, $c \equiv \boldsymbol{h}^T\boldsymbol{y}$, and $\boldsymbol{D} \equiv \boldsymbol{A}\boldsymbol{W}^{-1}$, then with a bit of algebraic manipulation we obtain:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} \;\rightarrow\; \boldsymbol{h}^T\boldsymbol{y} = \boldsymbol{h}^T\boldsymbol{A}\boldsymbol{x} \tag{2.71}$$

$$c = \boldsymbol{h}^T\boldsymbol{A}\boldsymbol{I}\boldsymbol{x} \tag{2.72}$$

$$c = \boldsymbol{h}^T\boldsymbol{A}(\boldsymbol{W}^{-1}\boldsymbol{W})\boldsymbol{x} \tag{2.73}$$

$$c = (\boldsymbol{h}^T\boldsymbol{A})\boldsymbol{W}^{-1}(\boldsymbol{W}\boldsymbol{x}) \tag{2.74}$$

$$c = (\boldsymbol{w}^T\boldsymbol{W}^{-1})\boldsymbol{z} \tag{2.75}$$

$$c = \boldsymbol{1}^T\boldsymbol{z}. \tag{2.76}$$

And similarly $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} = (\boldsymbol{A}\boldsymbol{W}^{-1})(\boldsymbol{W}\boldsymbol{x}) = \boldsymbol{D}\boldsymbol{z}$.

Thus problem $\left(P_0^+\right)$ is rewritten equivalently as:

$$\left(P_0^+\right): \quad \hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\text{argmin}}\, \|\boldsymbol{z}\|_0 \quad \text{s.t.: } \boldsymbol{D}\boldsymbol{z} = \boldsymbol{y}, \quad \boldsymbol{z} \geq 0, \quad \boldsymbol{1}^T\boldsymbol{z} = c. \tag{2.77}$$

Note that imposing the ANC constraint on the original problem enforces the constraint $\boldsymbol{1}^T\boldsymbol{z} = c$ which is strongly connected to the ASC constraint. In real images the variability of signatures introduces positive scaling factors varying from pixel to pixel. Thus the ASC should be replaced with a *generalized* ASC of the form $\sum_i \xi_i x_i = 1$, where the weights ξ_i are the scale factors varying in each pixel. Since the nonnegativity constraint implies the sum-to-one constraint indirectly, then we don't actually have to explicitly impose the ASC constraint in (2.77).

Problem $\left(P_0^+\right)$ is similar to (P_0), meaning that it is NP hard, nonconvex and impossible to solve for a general matrix $\boldsymbol{A}$. Thus we consider l_1-norm relaxation [9], Candes's miracle of higher dimensional geometry:

$$\left(P_1^+\right): \quad \hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\text{argmin}}\, \|\boldsymbol{z}\|_1 \quad \text{s.t.: } \boldsymbol{D}\boldsymbol{z} = \boldsymbol{y}, \quad \boldsymbol{z} \geq 0. \tag{2.78}$$

As above, with condition $\gamma_{2s} < 4\sqrt{2} - 3 \approx 2.6569$ applied to matrix $\boldsymbol{D}$ in problem $\left(P_1^+\right)$ this ensures that any s-sparse solution of $\left(P_0^+\right)$ is recovered by solving problem $\left(P_1^+\right)$ via the BP algorithm.

2.4.7 Approximate Solutions in the Presence of Noise

Reconsidering the original problem (P_0) in the presence of noise perturbation $\boldsymbol{n}$, we tolerate noise by amount δ to obtain:

$$\left(P_0^{\delta}\right): \quad \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_0 \quad \text{s.t.: } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \delta. \tag{2.79}$$

The concept of uniqueness is now replaced by the concept of *stability* [9], where a sparse vector solution $\boldsymbol{x}_0$ satisfying the sparsity constraint $\boldsymbol{x}_0 < (1 + 1/\mu(\boldsymbol{A}))/2$ such that $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \delta$, implies that every solution $\boldsymbol{x}_0^{\delta}$ of problem $\left(P_0^{\delta}\right)$ satisfies the stability condition:

$$\|\boldsymbol{x}_0^{\delta} - \boldsymbol{x}_0\|^2 \leq \frac{4\delta^2}{1 - \mu(\boldsymbol{A})(2\boldsymbol{x}_0 - 1)}. \tag{2.80}$$

As before we consider two relaxed strategies for computing the sparse solution of $\left(P_0^{\delta}\right)$.

Similarly to (P_0), problem $\left(P_0^{\delta}\right)$ is NP hard. This can be approached via the greedy OMP algorithm with stopping criteria $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \delta$. It is more efficient to use the BP algorithm by relaxing the l_0-norm obtaining the convex *basis pursuit denoising* (BPDN) optimization problem [91]:

$$\left(P_1^{\delta}\right): \quad \hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_1 \quad \text{s.t.: } \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \leq \delta. \tag{2.81}$$

The solution to $\left(P_1^{\delta}\right)$ is computed using convex optimization methods, where the sparsity and stability is provided by the condition $\|\boldsymbol{x}_1^{\delta} - \boldsymbol{x}\|_2 \leq \delta$, and δ depends on the restricted isometric constants α_{2s} and β_{2s} satisfying $\gamma_{2s} < 4\sqrt{2} - 3 \approx 2.6569$.

As a *nonnegative signals* problem in the presence of noise, we can rewrite the problem in the equivalent nonnegative form, similar to the $\left(P_0^{+}\right)$ with the terms transformed as above in (2.77):

$$\left(P_0^{\delta+}\right): \quad \hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \|\boldsymbol{z}\|_0 \quad \text{s.t.:} \quad \|\boldsymbol{D}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \delta, \quad \boldsymbol{z} \geq 0, \tag{2.82}$$

where vector $\boldsymbol{h}$ is chosen such that the nonnegative condition $\boldsymbol{h}^T\boldsymbol{A} > 0$ holds. Defining matrix $\boldsymbol{D} \equiv \boldsymbol{A}\boldsymbol{W}^{-1}$, where $\boldsymbol{W} \equiv \operatorname{diag}(\boldsymbol{h}^T\boldsymbol{A})$. From linear system $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{z} + \boldsymbol{n}$ we obtain the l_2-ball condition $\|\boldsymbol{n}\|_2^2 = \|\boldsymbol{D}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \delta$, and we can rewrite the previous summation constraint as $\mathbf{1}^T\boldsymbol{z} = c + \boldsymbol{h}^T\boldsymbol{n}$. Thus the nonnegativity constraint imposes a soft constraint $\|\boldsymbol{h}^T\boldsymbol{n}\|_2^2 = \|\mathbf{1}^T\boldsymbol{z} - c\|_2^2 \leq \delta_h$ since $\|\boldsymbol{n}\|_2^2 \leq \delta$.

Similar to problem $\left(P_0^{\delta}\right)$, nonconvex problem $\left(P_0^{\delta+}\right)$ is NP hard and impossible to solve exactly for matrix $\boldsymbol{A}$ or nonnegative $\boldsymbol{D}$. As before we consider l_1-norm relaxation:

$$\left(P_1^{\delta+}\right): \quad \hat{\boldsymbol{z}} = \underset{\boldsymbol{z}}{\operatorname{argmin}} \|\boldsymbol{z}\|_1 \quad \text{s.t.:} \quad \|\boldsymbol{D}\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \delta, \quad \boldsymbol{z} \geq 0, \tag{2.83}$$

which also has to satisfy sparsity condition $\gamma_{2s} < 4\sqrt{2} - 3 \approx 2.6569$ applied to the restricted isometric constants of matrix $\boldsymbol{D}$, which ensures stability of convex problem $\left(P_1^{\delta+}\right)$.

2.4.8 Sparse Unmixing by Variable Splitting and Augmented Lagrangian

All the problems in the previous section be rewritten in their equivalent Lagrangian formulation, [100]. For example, problem $\left(P_1^{\delta+}\right)$ can be rewritten for an appropriate choice of regularization parameter λ as follows:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \quad \text{s.t.:} \quad \boldsymbol{x} \succeq \boldsymbol{0} \tag{2.84}$$

Even though problem (2.84) is convex, it is non-differentiable and cannot be easily solved, [18]. The ASC forces the non-negative abundance coefficients to have a constant sum. Thus the l_1-norm for abundance vector $\boldsymbol{x}$ is constant and cannot be minimized without breaking the ASC.

This problem can be solved efficiently without the ASC using proximal methods [101], such as the *alternating direction method of multipliers* (ADMM), [102]. The ADMM proximal methods are implemented in the *Sparse Unmixing by Variable Splitting and Augmented Lagrangian* (SUnSAL) algorithm by Bioucas-Dias [98], and his computation of the ADMM's solution, presented in the next chapter, is based on Eckstein and Bertsekas's theorem 1 stating the convergence of ADMM, [103].

Eckstein and Bertsekas consider the following unconstrained problem:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} f_1(\boldsymbol{x}) + f_2(\boldsymbol{G}\boldsymbol{x}) \tag{2.85}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $f_1 : \mathbb{R}^n \to \mathbb{R}$, $f_2 : \mathbb{R}^p \to \mathbb{R}$, and $\boldsymbol{G} \in \mathbb{R}^{p \times n}$. Their theorem states:

Theorem 1 (Eckstein and Bertsekas, 1992)**.** *Let $\boldsymbol{G}$ have full column rank and f_1, f_2 be closed, proper, and convex. Consider arbitrary $\mu > 0$ and $\boldsymbol{u}_0$, $\boldsymbol{d}_0 \in \mathbb{R}^p$. Consider three sequences $\{\boldsymbol{x}_k \in \mathbb{R}^n,\, k = 0, 1, \ldots\}$, $\{\boldsymbol{u}_k \in \mathbb{R}^p,\, k = 0, 1, \ldots\}$, and $\{\boldsymbol{d}_k \in \mathbb{R}^p,\, k = 0, 1, \ldots\}$ that satisfy*

$$\boldsymbol{x}_{k+1} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, f_1(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{G}\boldsymbol{x} - \boldsymbol{u}_k - \boldsymbol{d}_k\|_2^2 \tag{2.86}$$

$$\boldsymbol{u}_{k+1} = \underset{\boldsymbol{u}}{\operatorname{argmin}}\, f_2(\boldsymbol{u}) + \frac{\mu}{2}\|\boldsymbol{G}\boldsymbol{x}_{k+1} - \boldsymbol{u} - \boldsymbol{d}_k\|_2^2 \tag{2.87}$$

$$\boldsymbol{d}_{k+1} = \boldsymbol{d}_k - (\boldsymbol{G}\boldsymbol{x}_{k+1} - \boldsymbol{u}_{k+1}). \tag{2.88}$$

Then, if (2.85) *has a solution, the sequence* $\{\boldsymbol{x}_k\}$ *converges to it; otherwise, at least one of the sequences* $\{\boldsymbol{u}_k\}$ *or* $\{\boldsymbol{d}_k\}$ *diverges.*

The idea behind the SUnSAL algorithm is that it forces a certain number of rows of the abundance matrix $\boldsymbol{X}$ to be entirely zero, since there will be a lot of redundant spectral signatures in the library that will not be present in any pixel of the scene. The number of non-zero rows of the abundance matrix $\boldsymbol{X}$ has to be very small. The reformulated sparse unmixing problem for the *whole image* is as follows:

$$\hat{\boldsymbol{X}} = \underset{\boldsymbol{X}}{\operatorname{argmin}}\, \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \lambda \sum_{i=1}^{m} \|\boldsymbol{x}_i\|_2 \tag{2.89}$$

where $\boldsymbol{Y} \in \mathbb{R}^{B\times N}$ is the matrix of observed data, the library $\boldsymbol{A} \in \mathbb{R}^{B\times m}$, and the abundance matrix $\boldsymbol{X} \in \mathbb{R}^{m\times N}$. Also we mentioned in section 2.2.1 that the Frobenius norm is more efficient to compute for matrices rather than the natural norms.

The term $\sum_{i=1}^{m} \|\boldsymbol{x}_i\|_2$ can be seen as the l_1-norm of a m-dimensional vector containinig in each entry the l_2-norm of the abundance coefficients for a pixel. This penalty term can be interpreted as the mixed $l_{p,q}$-norm as defined in equation (2.9), section 2.2.1, giving us the "collaborative" sparsity penalty $l_{2,1}$-norm:

$$\|\boldsymbol{X}\|_{2,1} = \sum_{i=1}^{m} \left(\sum_{j=1}^{N} |x_{ij}|^2 \right)^{1/2} \tag{2.90}$$

which is basically the sum of Euclidean norms of abundance matrix columns. This helps us rewrite the optimization problem for the SUnSAL algorithm as follows:

$$\hat{\boldsymbol{X}} = \underset{\boldsymbol{X}}{\operatorname{argmin}}\, \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_{2,1} \tag{2.91}$$

The SUnSAL generates piecewise smooth abundance maps, and with $SNR > 20\,dB$ we obtain visually meaningful abundance maps, [18].

2.5 Summary of Theoretical Background

In this chapter we have presented a broad introduction to image spectroscopy, the linear mixing model, the hyperspectral unmixing processing chain, and focused on geometrical unmixing methods to extract endmembers, and sparse unmixing methods which solve the inverse problem without the need for endmember extraction. The following are the main points to be digested from this chapter:

- Atmospheric correction of the raw radiance is necessary, as spectral libraries collected on the ground will not match remotely sensed data exactly.
- Dimension reduction is beneficial in terms of efficient computation and data storage. It also makes it easier to process data projected on the lower-dimensional subspace.
- The LMM finds limited use for images where pixels are intimately mixed or there are obstructions on the ground. However the LMM does well for most real scenes in practice.
- Geometrical endmember extraction algorithms are prone to erroneous endmember estimation, however these algorithms are still in common use in the industry. Sparse coding and the increased availability of spectral libraries from USGS and NASA, will probably impact the use of geometrical endmember extraction algorithms in the near future.
- FCLS is the common method of solving the inverse problem, however it does not account for sparse solutions. Thus sparse unmixing methods such as SUnSAL are becoming increasingly popular.

Chapter 3

Computer Aided Modeling

In the previous sections we listed a set of optimization problems aimed at computing exact and approximate solutions to the hyperspectral SR unmixing problem, which all depend on the availability of spectral libraries. This research report presents sparse unmixing algorithms in detail. Each of the SR algorithms addresses specific problems specified above.

To illustrate these concepts briefly, the popular *greedy* OMP algorithm is tailored to l_0-norm minimization problems such as $\left(P_0^+\right)$ and $\left(P_0^{\delta+}\right)$. The BP and BPDN algorithms introduced by Bioucas-Dias et al [98], compute approximations to the l_1-norm minimization problems, which are *constrained sparse unmixing* algorithms via variable *splitting and augmented Lagrangian* (CSUnSAL). Thus the CSUnSAL algorithm variations are aimed at linear problems (P_1) and $\left(P_1^+\right)$, and quadratic problems $\left(P_1^\delta\right)$ and $\left(P_1^{\delta+}\right)$, which are tailored to hyperspectral applications with hundreds of thousands or even millions of spectral vectors to unmix [9]. As for *unconstrained* sparse unmixing, BP and BPDN algorithms fall under the SUnSAL algorithms category. All constrained sparse unmixing problems (P_1), $\left(P_1^+\right)$, $\left(P_1^\delta\right)$ and $\left(P_1^{\delta+}\right)$ can be converted to equivalent unconstrained versions by minimizing the respective Lagrangian. These become *least squares* (LS) models with l_1-norm regularizers, for example problem $\left(P_1^\delta\right)$ can be rewritten in its equivalent form as:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1. \tag{3.1}$$

Iordache also makes use of the *iterative spectral mixture analysis* (ISMA) to solve the unconstrained spectral unmixing problem (2.62). ISMA [104] is an iterative method which finds the optimal endmember set by analyzing the change in the *root-mean-square error* (rmse) after reconstructing the original scene using estimated fractional abundances. Bioucas-Dias also proposed the *two-step iterative shrinkage / thresholding* (TwIST) algorithm [105] which also solves the unconstrained optimization problem iteratively, combining the linear mixture model with a nonquadratic

regularizer. We will not be considering ISMA and TwIST in this report.

Improvements to these types of algorithms have been made by efficiently pruning the spectral libraries [106], spatial-spectral unmixing with total variance (TV) regularizes [107], parallel high performance computing [108], and many more [109].

In this research report, we only consider geometrically-based endmember extraction algorithms such as PPI, N-FINDR, SISAL and VCA, and the sparse regression algorithms OMP and SUnSAL (based on ADMM).

3.1 Data Simulation

Firstly we select a spectral library A from the following choices:

1. $\boldsymbol{A}_1 = \texttt{randn}(B, m)$, randomly generated normal i.i.d values;
2. $\boldsymbol{A}_2 = \texttt{rand}(B, m)$, randomly generated uniformly i.i.d. values;
3. $\boldsymbol{A}_3 =$ USGS 1995 spectral library;
4. $\boldsymbol{A}_4 =$ USGS 1995 spectral library pruned by 3°;
5. $\boldsymbol{A}_5 =$ artificial Cuprite image to contain only k endmembers;
6. $\boldsymbol{A}_6 =$ artificial terrain image to contain only k endmembers;

where "i.i.d." stands for *independent and identically distributed*, B is the number of bands, m is the number of materials. Function `randn`() draws from the Gaussian distribution, while function `rand`() draws from uniform distribution. We keep the naming convention of the majority of these function as they are named in MATLAB.

The USGS libraries were downloaded from Bioucas-Dias's hyperspectral unmixing overview available on his publications web page by searching for paper [5] and clicking the "MATLAB code" link. José Bioucas-Dias has pruned the USGS 1995 library to contain only endmembers with a specified angle difference between, ensuring some independence between the spectral signatures. The Cuprite and terrain images are available in the same "datasets" folder from Bioucas-Dias's publication [5]. We performed endmember extraction using the N-FINDR algorithm and only selected k endmembers.

Algorithm 1 Simulating data

Inputs:

$\boldsymbol{A}_s$ is the selected library;

$k \in \{2, 3, 4, \ldots, 20\}$ is the sparsity, i.e. number of endmembers present in a scene;

$SNR \in \{20, 30, 40, 50\}$ given in decibels (dB);

N is the number of pixels in image, i.e. number of rows $\times$ number of columns;

B is the number of spectral bands;

m is the number of materials in library;

$toy \in \{\text{true}, \text{false}\}$ i.e. whether to permute k endmembers;

s is the library choice selected;

Initialize:

$\boldsymbol{X} \leftarrow \boldsymbol{0}_{m \times N}$, ie. $m \times N$ matrix of zeros;

if $s \in \{6, 7\}$ **then**

$\boldsymbol{Y} \leftarrow \texttt{reshape}(\boldsymbol{I}_A, N, B)$, where $\boldsymbol{I}_A$ is the 3D artificial image, i.e. $\boldsymbol{Y} \in \mathbb{R}^{N \times B}$;

$\boldsymbol{X} \leftarrow \texttt{reshape}(\boldsymbol{X}_A, N, m)$, where $\boldsymbol{X}_A$ is the 3D abundance map, i.e. $\boldsymbol{X} \in \mathbb{R}^{N \times m}$;

else

if $toy ==$ true **then**

$indices \leftarrow \texttt{randperm}(m)$;

$sparse \leftarrow indices_{\{1:k\}}$, i.e. chose the same k endmembers for each pixel;

$\boldsymbol{X}_{\{sparse,:\}} \leftarrow \texttt{Dir}(\boldsymbol{1}_k, N)^T$, i.e. k sparse entries summing to 1, and 0's elsewhere;

else

for $i = 1$ to N **do**

$indices \leftarrow \texttt{randperm}(m)$;

$sparse \leftarrow indices_{\{1:k\}}$, different permutation for each pixel;

$\boldsymbol{X}_{\{sparse,1\}} \leftarrow \texttt{Dir}(\boldsymbol{1}_k, 1)^T$

end for

end if

end if

$\sigma \leftarrow \sqrt{\|\boldsymbol{A}_s \boldsymbol{X}\|_F^2 / (B \times N \times 10^{SNR/10})}$, noise standard deviation;

$\boldsymbol{\xi} \leftarrow \sigma \times \texttt{randn}$, generate noise matrix;

$\boldsymbol{Y} = \boldsymbol{A}_s \boldsymbol{X} + \boldsymbol{\xi}$, artificial data according to LMM (2.12) ;

where $\texttt{Dir}(\boldsymbol{A}, d)$ is a custom function written by Bioucas-Dias to draw random numbers from the Dirichlet distribution, where parameters $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, or $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ if dimension d is given. Each returned column sums to 1, hence the transpose.

The method of generating the abundance matrix is exactly according to the way Bioucas-Dias generates it. The addition of *toy* $\neq$ true scenario is my own contribution.

The noise calculation by Bioucas-Dias results from solving for $\boldsymbol{\xi}$ in the following equation, where SNR is given in decibels:

$$10^{SNR/10} = \|\boldsymbol{A}_s\boldsymbol{X}\|_F^2/\|\boldsymbol{\xi}\|_F^2 \tag{3.2}$$

where the Frobenius norm was give by equation (2.10) in section 2.2.1.

The artificial data $\boldsymbol{Y}$ generated in algorithm 1 uses the LMM (2.12) specified in chapter 2.2.3, with the selected library as input. The abundances map $\boldsymbol{X}$ comes in handy for validating unmixing results.

3.2 Algorithms

This section provides the pseudo code for unmixing algorithms used in this report. The majority of these algorithms have been implemented by Bioucas-Dias for MATLAB, all code available at http://www.lx.it.pt/ bioucas/code.htm based on his publications. The other authors have been credited in the following sections. The literature background for each of the following algorithm has already been provided in chapter 2. Supplementary details and explanation have been provided below where necessary.

3.2.1 Dimension Reduction

Signal subspace identification is an important first step in hyperspectral unmixing as it enables correct dimensionality reduction, improves computational performance and lowers complexity. Bioucas-Dias and Nascimento's method of subspace identification employs a minimum mean-square error approach, which is based on eigendecomposition, it is unsupervised and fully-automatic. [110].

There are three steps implemented by Bioucas-Dias in MATLAB: (i) estimate the noise present in the signal; (ii) identify the lower-dimensional subspace to project the signal on; and (iii) carry out an affine projection to ensure the projected signal

does not suffer from rotation or scaling effects.

Signal Noise Estimation

First we estimate the noise matrix. The inputs are the 2-D signal $\boldsymbol{Y} \in \mathbb{R}^{B \times N}$, B is the number of spectral bands, and N is the total number of pixels.

Algorithm 2 Noise Estimation [110]

1: **Inputs:**
   $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]$, where $\boldsymbol{Y} \in \mathbb{R}^{B \times N}$;
2: **Initialize:**
   $\boldsymbol{Z} \leftarrow \boldsymbol{Y}^T$;
   $\hat{\boldsymbol{R}} \leftarrow \left(\boldsymbol{Z}^T \boldsymbol{Z}\right)$;
   $\boldsymbol{R}' \leftarrow \hat{\boldsymbol{R}}^{-1}$;
3: **for** $i = 1$ to B **do**
4: $\quad \hat{\boldsymbol{\beta}}_i \leftarrow ([\boldsymbol{R}']_{\partial_i,\partial_i} - [\boldsymbol{R}']_{\partial_i,i}[\boldsymbol{R}']_{i,\partial_i}/[\boldsymbol{R}']_{i,i})\,[\hat{\boldsymbol{R}}]_{\partial_i,i}$, regression step;
5: $\quad \hat{\boldsymbol{\xi}}_i \leftarrow \boldsymbol{z}_i - \boldsymbol{Z}_{\partial_i}\hat{\boldsymbol{\beta}}_i$, noise estimation;
6: **end for**
7: **return** $\hat{\boldsymbol{\xi}}$.

The output noise matrix of algorithm 2 is $\hat{\boldsymbol{\xi}} \in \mathbb{R}^{N \times B}$, whereas the symbol $[\hat{\boldsymbol{R}}]_{\partial_i,\partial_i}$ denotes the matrix obtained from $\hat{\boldsymbol{R}} \leftarrow \left(\boldsymbol{Z}^T \boldsymbol{Z}\right)$ by deleting the ith row and i^{th} column. $[\hat{\boldsymbol{R}}]_{i,\partial_i}$ denotes the ith row of $[\hat{\boldsymbol{R}}]_{:,\partial_i}$ which has i^{th} column deleted, and vice versa for $[\hat{\boldsymbol{R}}]_{\partial_i,i}$. Note the iterations are computed for each spectral band for $i = 1, \ldots, B$, where B is the number of spectral bands.

Subspace Identification

Bioucas-Dias [110] makes frequent use of the *hyperspectral signal subspace identification by minimum error* (HySime) algorithm to identify the lower-dimensional subspace, by selecting a subset of eigenvectors that best represents the signal subspace in the least squared error sense. This is achieved by optimizing the cost function (3.3):

$$\left(\hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\pi}}\right) = \underset{\boldsymbol{\delta},\boldsymbol{\pi}}{\operatorname{argmin}} \left\{ c + \sum_{j=1}^{k} \underbrace{-\boldsymbol{e}_{i_j}^T \hat{\boldsymbol{R}}_y \boldsymbol{e}_{i_j} + 2\boldsymbol{e}_{i_j}^T \hat{\boldsymbol{R}}_n \boldsymbol{e}_{i_j}}_{\delta_{i_j}} \right\} \tag{3.3}$$

where $\hat{\boldsymbol{R}}_y$ and $\hat{\boldsymbol{R}}_n$ are the correlation matrices of the signal and noise, and $\boldsymbol{e}$'s are the eigenvectors of the estimated correlation matrix $\hat{\boldsymbol{R}}_x$. The first variable op-

timized is $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_B]$ which has elements $\delta_i = \sum_j \left(-\boldsymbol{e}_{i_j}^T \hat{\boldsymbol{R}}_y \boldsymbol{e}_{i_j} + 2\boldsymbol{e}_{i_j}^T \hat{\boldsymbol{R}}_n \boldsymbol{e}_{i_j} \right)$, and $i = 1, \ldots, B$. The summation in (3.3) is performed for k elements, the number of eigenvectors from the eigen-decomposition of $\hat{\boldsymbol{R}}_x$. Sorting the cost vector $\boldsymbol{\delta}$ in ascending order we store the optimal permutation in $\boldsymbol{\pi}$.

Algorithm 3 HySime [110]

Inputs:

$\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]$ and $\boldsymbol{Y}^T \in \mathbb{R}^{N \times B}$;

$\hat{\boldsymbol{\xi}}$ the $B \times N$ estimated noise matrix from Algorithm 2;

Initialize:

$\hat{\boldsymbol{R}}_y \leftarrow \left(\boldsymbol{Y}\boldsymbol{Y}^T \right) / N$;

$\hat{\boldsymbol{R}}_n \leftarrow 1/N \sum_i \left(\hat{\boldsymbol{\xi}}_i \hat{\boldsymbol{\xi}}_i^T \right)$;

$\hat{\boldsymbol{R}}_x \leftarrow 1/N \sum_i \left(\left(\boldsymbol{y}_i - \hat{\boldsymbol{\xi}}_i \right) \left(\boldsymbol{y}_i - \hat{\boldsymbol{\xi}}_i \right)^T \right)$;

$\boldsymbol{E} \leftarrow [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_B]$ eigenvectors given by $\texttt{svd}\left(\hat{\boldsymbol{R}}_x \right)$;

$\boldsymbol{\delta} \leftarrow [\delta_1, \ldots, \delta_B]$ where $\delta_i = \sum_{j=1}^{k} \left(-\boldsymbol{e}_{i_j}^T \hat{\boldsymbol{R}}_y \boldsymbol{e}_{i_j} + 2\boldsymbol{e}_{i_j}^T \hat{\boldsymbol{R}}_n \boldsymbol{e}_{i_j} \right)$ in (3.3);

$\left(\hat{\boldsymbol{\delta}}, \hat{\boldsymbol{\pi}} \right) = \text{Sort}\left(\boldsymbol{\delta} \right)$, sort δ_i ascending, save permutation $\hat{\pi}$;

$\hat{k} \leftarrow$ number of terms $\hat{\delta}_i < 0$;

return $\hat{\boldsymbol{E}} = \left[\boldsymbol{e}_{\hat{i}_1}, \ldots, \boldsymbol{e}_{\hat{i}_{\hat{k}}} \right] \in \mathbb{R}^{N \times \hat{k}}$ and $\hat{k}$ is the sparsity estimate.

The inputs are the spectral data and the noise estimate matrix from algorithm 2. The sample correlation matrix $\hat{\boldsymbol{R}}_y$, the noise correlation matrix $\hat{\boldsymbol{R}}_n$, and the signal correlation matrix $\hat{\boldsymbol{R}}_x$ are calculated next. The eigenvectors of the signal correlation matrix are calculated next by function `svd()`, which in turn allows for calculating terms δ_i based on the quadratic cost function (3.3). The last steps of algorithm 3 provide the minimisation of the cost function, and retrieve the signal subspace $\hat{\boldsymbol{E}}$ from $\hat{k}$ and $\hat{\boldsymbol{\pi}}$.

Basis matrix $\hat{\boldsymbol{E}}$ is used to project the hyperspectral data onto the signal subspace: $\boldsymbol{Y}_e^T = \hat{\boldsymbol{E}}\hat{\boldsymbol{E}}^T\boldsymbol{Y}^T \in \mathbb{R}^{N \times \hat{k}}$. Projected data $\boldsymbol{Y}_e$ is input into the *affine projection* algorithm next, to ensure that the sum-to-one constraint holds.

Affine Projection

The input signal is usually the signal projected on the lower-dimensional subspace $\boldsymbol{Y}_e^T = \hat{\boldsymbol{E}}\hat{\boldsymbol{E}}^T\boldsymbol{Y}^T$. Instead of the input the raw signal $\boldsymbol{Y} \in \mathbb{R}^{B \times N}$, we use $\boldsymbol{Y}$ to denote the lower dimensional $\boldsymbol{Y}_e \in \mathbb{R}^{p \times N}$. Bioucas-Dias mainly uses the projected signal

$\boldsymbol{Y}_e$. Also see section 2.3.2 for more details.

Algorithm 4 Affine Projection [62], implemented in [5]

Inputs:

$\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N] \in \mathbb{R}^{B \times N}$;

$\hat{k}$ the sparsity estimate from algorithm 3;

projectionType $\in$ { "orthogonal", "DPFT" }

Initialize:

$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_B]^T$ where $\mu_i \leftarrow 1/N \sum_j y_{ij}$, the mean component;

$\boldsymbol{u} \leftarrow \mathbf{1} / \left(\mathbf{1}^T \boldsymbol{\mu}\right)$, where $\mathbf{1}$ is a $B \times 1$ vector of ones;

$p \leftarrow \hat{k}$;

if *projectionType* == "orthogonal" **then**

$\boldsymbol{M} = [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_N]$ where $\boldsymbol{\mu}_j = \boldsymbol{\mu}$ for each column j;

$\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{Y} - \boldsymbol{M}$;

$\hat{\boldsymbol{U}}_p \leftarrow \texttt{svds}(\hat{\boldsymbol{Y}}\hat{\boldsymbol{Y}}^T/N, p-1)$, see `svds` explanation below;

$\boldsymbol{Y}'_p \leftarrow \hat{\boldsymbol{U}}_p^T \hat{\boldsymbol{Y}}$, project signal onto $\mathbb{R}^{p-1}$ subspace;

$\boldsymbol{\mu}_\perp \leftarrow \boldsymbol{\mu} - \hat{\boldsymbol{U}}_p \hat{\boldsymbol{U}}_p^T \boldsymbol{\mu}$, orthogonal components;

$\boldsymbol{U}_p \leftarrow \left[\hat{\boldsymbol{U}}_p, \boldsymbol{\mu}_\perp / \sqrt{\sum_{i=1}^{B} {\boldsymbol{\mu}_\perp}_i^2}\right]$, p orthogonal directions;

$\boldsymbol{\mu}'_\perp \leftarrow \boldsymbol{U}_p \boldsymbol{\mu}$, coordinates with respect to $\boldsymbol{U}_p$;

$\boldsymbol{M}' = [\boldsymbol{\mu}'_{\perp_1}, \ldots, \boldsymbol{\mu}'_{\perp_N}]$ where $\boldsymbol{\mu}'_{\perp_j} = \boldsymbol{\mu}'_\perp$ for each column j;

$\boldsymbol{Y}_p \leftarrow \boldsymbol{Y}'_p + \boldsymbol{M}'$, lift up to $\mathbb{R}^p$ subspace;

else[*projectionType* == "DPFT"]

$\boldsymbol{U}_p \leftarrow \texttt{svds}(\boldsymbol{Y}\boldsymbol{Y}^T/N, p)$;

$\hat{\boldsymbol{Y}}_p \leftarrow \boldsymbol{U}_p^T \boldsymbol{Y}$, project signal onto $\mathbb{R}^p$ subspace;

$\hat{u} \leftarrow \left(\boldsymbol{U}_p^T \boldsymbol{u}\right)^T \hat{\boldsymbol{Y}}_p$, perspective projection scaling factor;

$\boldsymbol{Y}_p = [\boldsymbol{y}_{p_1}, \ldots, \boldsymbol{y}_{p_N}]$, scaling $\boldsymbol{y}_{p_j} \leftarrow \hat{\boldsymbol{y}}_{p_j}/\hat{u}$ for each column j;

end if

return $\boldsymbol{Y}_p$, $\boldsymbol{U}_p$.

If we use the projected matrix $\boldsymbol{Y}_e^T = \hat{\boldsymbol{E}}\hat{\boldsymbol{E}}^T\boldsymbol{Y}^T \in \mathbb{R}^{N \times \hat{k}}$, then instead of spectral dimension of B, we have the lower $\hat{k}$. Algorithm 4 also mentions the function $\texttt{svds}(\boldsymbol{A}, p)$, which is MATLAB's *singular value decomposition* and it returns only the p largest singular values.

The projections onto a hyperplane, [5], are either orthogonal projections which suffer rotation effects, or perspective (DPFT) projection which are affected by scaling factor $\boldsymbol{y}/(\boldsymbol{y}^T\boldsymbol{u})$. The hyperplane in figure 3.1 is defined by $\boldsymbol{y}^T\boldsymbol{u} = 1$, and vector

$\boldsymbol{u}$ is chosen such that $\boldsymbol{y}^T\boldsymbol{u} > 0$. Affine projection does fix projection effects such as rotation and scaling, and ensures the sum-to-one constraint.

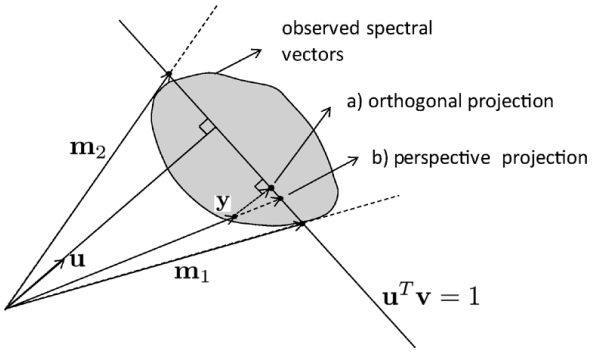


Fig. 3.1: Orthogonal projection and DPFT projection, source: Bioucas-Dias [5]

3.2.2 Geometric Endmember Extraction Algorithms

In these classic unmixing algorithms one can input the original signal $\boldsymbol{Y} \in \mathbb{R}^{B\times N}$, however Bioucas-Dias inputs the affinely projected signal $\boldsymbol{Y} \in \mathbb{R}^{p\times N}$. These algorithms have been discussed in detail in section 2.3.3.

Pure Pixel Based Algorithms

The *pixel purity index* (PPI) algorithm developed by Boardman in [111],[112], has been widely used in hyperspectral image analysis for endmember extraction mainly due to its availability in the *ENvironment for Visualizing Images* (ENVI) software.

The PPI algorithm has a supervised nature [63], and requires the virtual dimensionality to be given, i.e. the number of endmembers to extract. Noise-whitening and dimensionality reduction are normally applied prior using MNF or PCA, see section 2.3.1. Bioucas-Dias applies HySime and affine projection instead.

A pixel purity score is calculated for each pixel in the data cube. Using a random generator we produce a sufficiently large set of "skewers" to project the sample data onto. All the samples are projected onto the skewers, and the ones falling at the extremes of the skewers are counted. This process is repeated many times, and the pixels with the number of votes above a cut-off threshold are considered "pure".

This is the very reason why we call these "pure" pixels the "endmembers", because that sit at the ends of the skewers.

Algorithm 5 Pixel Purity Index [111], implemented by Greg Isaac

Inputs:

$\boldsymbol{Y} \in \mathbb{R}^{p \times N}$;

q is the number of endmembers to extract;

s is the number of "skewer" vectors to project data on;

Initialize:

$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_B]^T$ where $\mu_i \leftarrow 1/N \sum_j y_{ij}$;

$\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N]$, where $\hat{\boldsymbol{y}}_j \leftarrow \boldsymbol{y}_j - \boldsymbol{\mu}$;

$\boldsymbol{S} \leftarrow \texttt{randn}(p, s)$, generate normally distributed skewers;

$\boldsymbol{v} \leftarrow \boldsymbol{0}$, $N \times 1$ vector of votes;

for $k = 1$ to q **do**

$\boldsymbol{Y}^* \leftarrow \left| \boldsymbol{s}_k^T \hat{\boldsymbol{Y}} \right|$, project all data onto skewer $\boldsymbol{s}_k \in \boldsymbol{S}$;

$\hat{j} \leftarrow \texttt{max}(\boldsymbol{Y}^*_{:,j})$, index of max correlation;

$\boldsymbol{v}_{\hat{j}} \leftarrow \boldsymbol{v}_{\hat{j}} + 1$, update votes;

end for

Sort votes $\boldsymbol{v}$ in descending order of votes;

$\hat{\boldsymbol{v}} \leftarrow$ subset indices for the highest q votes;

$\boldsymbol{M} \leftarrow \boldsymbol{Y}_{:,\hat{\boldsymbol{v}}}$ subset of original matrix with highest votes;

return $\boldsymbol{M}$.

The output matrix $\boldsymbol{M}$ contains only the signatures for the pixels assumed to be pure, i.e. the material signatures. However this is not always valid in practice.

Near Pure Pixel Based Algorithms

A more realistic approach to geometrically unmixing hyperspectral data is Winter's popular N-FINDR algorithm, [54], [64], which evaluates the likelihood of each pixel being pure or *nearly* pure by "inflating" the volume of a simplex inside the data and evaluates if each pixel is located on a vertex of simplex or near the extrema.

N-FINDR needs to start off with a MNF transformation, and requires to know the number of endmembers to be extracted in order to build the simplex volume encapsulating the data cloud. The simplex volume is refined at every iteration, by calculating a trial volume by trying to place each pixel in all the endmember positions, until an optimal volume is found.

Algorithm 6 N-FINDR [54], implemented in [5]

Inputs:

$\boldsymbol{Y} \in \mathbb{R}^{B \times N}$;

p the number of endmembers to find;

Initialize:

$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_B]^T$ where $\mu_i \leftarrow 1/N \sum_j y_{ij}$;

$\boldsymbol{R} \leftarrow \boldsymbol{Y}\boldsymbol{Y}^T/N$, correlation matrix;

$\boldsymbol{\Sigma} \leftarrow \boldsymbol{R} - \boldsymbol{\mu}\boldsymbol{\mu}^T$, covariance;

$\boldsymbol{U} \leftarrow \texttt{svds}(\boldsymbol{\Sigma}, p-1)$, orthogonal basis;

$\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N]$ where $\hat{\boldsymbol{y}}_j = \boldsymbol{y}_1 - \boldsymbol{\mu}$ for each j;

$\boldsymbol{Y}_{p-1} \leftarrow \boldsymbol{U}^T\hat{\boldsymbol{Y}}$, project zero-mean data onto $(p-1)$ subspace;

$\boldsymbol{s} \leftarrow p$-randomly selected indices out of N permuations;

$\boldsymbol{Y}_p = [\boldsymbol{1}; \boldsymbol{Y}_{p-1}]$, where $\boldsymbol{1}$ is a $1 \times N$ vector of ones, $\boldsymbol{Y}_p \in \mathbb{R}^{p \times N}$;

$\boldsymbol{M} \leftarrow$ subset of $\boldsymbol{Y}_p$ with indices in set $\boldsymbol{s}$, and $\boldsymbol{M} \in \mathbb{R}^{p \times p}$;

$v \leftarrow 0$, initialise volume;

for $i = p+1$ to N **do**

$\boldsymbol{y} \leftarrow$ random column from $\boldsymbol{Y}_p$, index not in subset $\boldsymbol{s}$;

for $j = 1$ to p **do**

$\boldsymbol{M}^* \leftarrow \boldsymbol{M}$;

$\boldsymbol{M}^*_{:,j} \leftarrow \boldsymbol{y}$, i.e. try pixel $\boldsymbol{y}$ in each endmember position;

$\boldsymbol{v}^*_j \leftarrow |\det\{\boldsymbol{M}^*\}|$;

end for

$\{v^*, \hat{j}\} \leftarrow \texttt{max}(\boldsymbol{v}^*)$, i.e. max value v^* and corresponding index $\hat{j}$;

if $v < v^*$ **then**

$\boldsymbol{M}_{:,\hat{j}} \leftarrow \boldsymbol{y}$, i.e. found an endmember;

$v \leftarrow v^*$;

update index set $\boldsymbol{s}$ with optimal position i;

end if

end for

$\hat{\boldsymbol{Y}} \leftarrow \boldsymbol{U}\boldsymbol{Y}_p$, project back to original dimensions;

$\hat{\boldsymbol{Y}} = [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_N]$, where $\hat{\boldsymbol{y}}_j \leftarrow \hat{\boldsymbol{y}}_j + \boldsymbol{\mu}$

$\hat{\boldsymbol{M}} \leftarrow$ subset $\hat{\boldsymbol{Y}}_{:,\boldsymbol{s}}$ with all updated indices;

return $\hat{\boldsymbol{M}}$.

We adopt Bioucas-Dias's methodology of inputting the dimension-reduced affinely projected $\boldsymbol{Y}_p \in \mathbb{R}^{p \times N}$ in N-FINDR. Alternatively, one can input the raw signal $\boldsymbol{Y} \in \mathbb{R}^{B \times N}$. The motivation for the dimension-reduced signal is more efficient computation. The output is the estimated materials matrix $\hat{\boldsymbol{M}}$. Next we consider Nascimento and Bioucas-Dias's *vertex component analysis* (VCA) algorithm [50].

Algorithm 7 Vertex Component Analysis [50], implemented in [5]

Inputs:

$\boldsymbol{Y} \in \mathbb{R}^{B \times N}$;

p the number of endmembers to find;

SNR the signal-to-noise ratio in decibels;

Initialize:

$\text{SNR}_{th} = 15 + 10 \log_{10}(p)\, dB$, threshold to decide projection;

if SNR $>$ SNR$_{th}$ **then**

$d \leftarrow p$, dimension for orthogonal projection;

$\boldsymbol{U}_d \leftarrow \texttt{svds}(\boldsymbol{Y}, d)$;

$\boldsymbol{X} \leftarrow \boldsymbol{U}_d^T \boldsymbol{Y}$, projection d-dimension subspace;

$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^T$ where $\mu_i \leftarrow 1/N \sum_j x_{ij}$;

$[\hat{\boldsymbol{Y}}]_{:,j} \leftarrow [\boldsymbol{X}]_{:,j} / ([\boldsymbol{X}]_{:,j}^T \boldsymbol{\mu})$, perspective projection;

else

$d \leftarrow p - 1$, dimension for orthogonal projection;

$\boldsymbol{R} \leftarrow \boldsymbol{Y}\boldsymbol{Y}^T / N$, correlation;

$\boldsymbol{\mu} = [\mu_1, \ldots, \mu_B]^T$ where $\mu_i \leftarrow 1/N \sum_j y_{ij}$;

$\boldsymbol{\Sigma} \leftarrow \boldsymbol{R} - \boldsymbol{\mu}^T \boldsymbol{\mu}$, covariance;

$\boldsymbol{U}_d \leftarrow \texttt{svds}(\boldsymbol{\Sigma}, d)$;

$[\boldsymbol{X}]_{:,j} \leftarrow \boldsymbol{U}_d^T (\boldsymbol{Y}_{:,j} - \boldsymbol{\mu})$, project the zero-mean data on subspace;

$c \leftarrow \text{argmax}_{j=1,\ldots,N} \|[\boldsymbol{X}]_{:,j}\|$

$\boldsymbol{c} = [c_1, c_2, \ldots, c_N]$, where $c_j \leftarrow c$ for each j;

$\hat{\boldsymbol{Y}} \leftarrow [\boldsymbol{X}; \boldsymbol{c}]$;

end if

$\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_p]$ is a $p \times p$ auxiliary matrix, where $\boldsymbol{a}_1 = [0, \ldots, 0, 1]^T$ and $\boldsymbol{a}_j = [0, \ldots, 0]^T$ for $j = 2, \ldots, p$;

for $i = 1$ to p **do**

$\boldsymbol{w} \leftarrow \texttt{randn}(0, \boldsymbol{I}_p)$, i.e. a zero-mean random Gaussian vector of cov. $\boldsymbol{I}_p$;

$\boldsymbol{f} \leftarrow (\boldsymbol{w} - \boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{w}) / \|\boldsymbol{w} - \boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{w}\|$, where $\boldsymbol{A}^\dagger$ is the pseudoinverse of $\boldsymbol{A}$, and vector f is orthonormal to the subspace spanned by $\boldsymbol{A}$;

$\boldsymbol{v} \leftarrow \boldsymbol{f}^T \hat{\boldsymbol{Y}}$;

$k \leftarrow \text{argmax}_j |\boldsymbol{v}|$, find projection extreme;

$[indices]_i \leftarrow k$, stores pixel index;

$[\boldsymbol{A}]_{:,i} \leftarrow [\hat{\boldsymbol{Y}}]_{:,k}$;

end for

if SNR $>$ SNR$_{th}$ **then**

$\hat{\boldsymbol{M}} \leftarrow \boldsymbol{U}_d [\boldsymbol{X}]_{:,indices}$;

else

$\hat{\boldsymbol{M}} \leftarrow \boldsymbol{U}_d [\boldsymbol{X}]_{:,indices} + \boldsymbol{\mu}$;

end if

return $\hat{\boldsymbol{M}}$ is the $B \times p$ estimated mixing matrix.

VCA assumes that endmembers are the vertices of a simplex, and that the affine transformation of a simplex is also a simplex, [50].This unsupervised algorithm iteratively projects data onto a direction orthogonal to the subspace spanned by endmembers already found. The newly found endmember should correspond to a vertex of the simplex being optimised. VCA performs better than PPI and comparable to N-FINDR, [50].

The first decision the VCA algorithm makes is if SNR $>$ SNR$_{th}$ in order to decide whether to project the data onto a subspace of dimension p or $p-1$. Next the projections avoid numerical erros.

The auxiliary matrix $\boldsymbol{A}$ stores the projection of the estimated endmembers. For each iteration, the vector $\boldsymbol{f}$ is orthonormal to the space spanned by the auxiliary matrix, and tries to find the extremes of projection $\boldsymbol{v} = \boldsymbol{f}^T\hat{\boldsymbol{Y}}$. Assuming that pure pixels are located on the vertices, then this projection gives the indices of the pure pixels. Their indices are stored, which are used in the end to select the columns of the materials matrix.

Minimum Volume

The *minimum volume simplex analysis* (MVSA) algorithm [62] first identifies the signal subspace, and then it applies perspective projection DPFT to obtain shifted materials matrix $\hat{\boldsymbol{M}}_0 \equiv [\boldsymbol{0}, \hat{\boldsymbol{M}}]$.
The algorithm iteratively changes each facet one at a time, while holding the other facets of the simplex fixed, such that the volume $V(\hat{\boldsymbol{M}}_0) = \left|\det(\hat{\boldsymbol{M}})\right|/p!$ is minimized, ensuring that all the spectral vectors belong to this simplex.

Bioucas-Dias presents the general MVSA algorithm in [5] as follows:

$$
\begin{aligned}
&\text{for } t = 1, \ldots, p-1 \\
&\quad \hat{\boldsymbol{M}}_0^{t+1} = \underset{\boldsymbol{M}_0}{\operatorname{argmin}}\, V(\boldsymbol{M}_0) \\
&\quad \text{s.t. : } \operatorname{facets}(\boldsymbol{M}_0) = \operatorname{facets}(\hat{\boldsymbol{M}}_0^t), \\
&\qquad\quad \text{except for facet } i = (t \bmod p) \\
&\quad \text{s.t. : } \hat{\boldsymbol{M}}^{-1}\boldsymbol{y}_i \succeq 0,\ \boldsymbol{1}_p^T\hat{\boldsymbol{M}}^{-1}\boldsymbol{y}_i = 1 \\
&\qquad\quad \text{for } i = 1, \ldots, N
\end{aligned}
$$

To make sure the spectral vectors are in the simplex, the ANC and ASC constraints must be obeyed respectively, i.e. $\hat{\boldsymbol{M}}^{-1}\boldsymbol{y}_i \succeq 0$ and $\mathbf{1}_p^T\hat{\boldsymbol{M}}^{-1}\boldsymbol{y}_i = 1$ for $i = 1, \ldots, p$, where the symbol $\succeq$ means "greater than component wise".

In SISAL [77], the positivity hard constraints are replaced by soft constraints controlled by regularization parameters. For large values of the regularization parameter, the hard constraint is recovered.

The advantages of SISAL over MVSA, PPI, N-FINDR, etc. are as follows: (i) robustness to outliers and noise, which improves greatly on MVSA; (ii) robustness to poor initialization for unbounded problems; (iii) handles large data problems better than classic algorithms, especially where the pure pixel assumption fails.

This algorithm's line of attack to solve hard non-convex optimization problems is to solve a sequence of non-smooth convex subproblems. Firstly, SISAL uses variable splitting to formulate the constraints, then applies the augmented Lagrangian technique, [113]. This approach implements a faster and simpler alternate minimization scheme.

For the linear mixing model with its constraints:

$$\boldsymbol{Y} = \boldsymbol{M}\boldsymbol{X}, \quad \boldsymbol{X} \succeq \mathbf{0}, \quad \mathbf{1}_p^T\boldsymbol{X} = 1 \tag{3.4}$$

where it is assumed the number of endmembers p has been estimated and the signal subspace has been identified beforehand. Thus the signal projected onto the subspace is denoted by $\boldsymbol{Y} \in \mathbb{R}^{p\times N}$, the mixing matrix by $\boldsymbol{M} \in \mathbb{R}^{p\times p}$, and the fractional abundances map by $\boldsymbol{X} \in \mathbb{R}^{p\times N}$.

Given signal $\boldsymbol{Y}$ and estimated number of endmembers p as the inputs, the SISAL algorithm aims to infer the mixing matrix $\boldsymbol{M}$, and thereafter the abundances matrix $\boldsymbol{X}$, by fitting a minimum volume simplex to the data subject to constraints $\boldsymbol{X} \succeq \mathbf{0}$ and $\mathbf{1}_p^T\boldsymbol{X} = 1$, based on Craig's seminal work [62].

Bioucas-Dias formulates the optimization problem for equation (3.4) as follows, since the volume defined by the columns of the square matrix $\boldsymbol{M}$ is proportional to $|\det\{\boldsymbol{M}\}|$, in order to formulate the subproblems:

$$\boldsymbol{M}^* = \underset{\boldsymbol{M}}{\operatorname{argmin}} |\det\{\boldsymbol{M}\}|, \quad \text{s.t.: } \boldsymbol{Q}\boldsymbol{Y} \succeq \mathbf{0}, \quad \mathbf{1}_p^T\boldsymbol{Q}\boldsymbol{Y} = \mathbf{1}_N^T \tag{3.5}$$

where $\boldsymbol{Q} = \boldsymbol{M}^{-1}$. And since $\det\{\boldsymbol{Q}\} = 1/\det\{\boldsymbol{M}\}$ and vice versa, problem (3.5) is replaced with:

$$\boldsymbol{Q}^* = \underset{\boldsymbol{Q}}{\operatorname{argmin}} - \log|\det\{\boldsymbol{Q}\}|, \quad \text{s.t.: } \boldsymbol{QY} \succeq \boldsymbol{0}, \quad \mathbf{1}_p^T \boldsymbol{QY} = \mathbf{1}_N^T \tag{3.6}$$

where $\operatorname{argmin}_{\boldsymbol{Q}} - \log|\det\{\boldsymbol{Q}\}|$ is equivalent to $\operatorname{argmax}_{\boldsymbol{Q}} \log|\det\{\boldsymbol{Q}\}|$, i.e. minimising the mixing matrix's volume is equivalent to maximising its inverse's volume. The constraints in problem (3.6) define a convex set only if matrix $\boldsymbol{Q}$ is symmetric and positive-definite, which doesn't frequently hold in practice, and thus it cannot find a global optima. The SISAL algorithm aims to find "good" sub-optimal solutions [77], therefore the sum-to-one constraint $\mathbf{1}_p^T \boldsymbol{QY} = \mathbf{1}_N^T$ is simplified by letting $\mathbf{1}_p^T \boldsymbol{Q} = \boldsymbol{a}^T$ and post-multiplying each side of the ASC by $\boldsymbol{Y}^T \left(\boldsymbol{YY}^T\right)^{-1}$ obtaining $\boldsymbol{a}^T = \mathbf{1}_N \boldsymbol{Y}^T \left(\boldsymbol{YY}^T\right)^{-1}$. This simplification is necessary because vector $\mathbf{1}_N^T$ does not belong to the null space of $\boldsymbol{Y}$, and thus Bioucas-Dias rewrites problem (3.6) as:

$$\boldsymbol{Q}^* = \underset{\boldsymbol{Q}}{\operatorname{argmin}} - \log|\det\{\boldsymbol{Q}\}|, \quad \text{s.t.: } \boldsymbol{QY} \succeq \boldsymbol{0}, \quad \mathbf{1}_p^T \boldsymbol{Q} = \boldsymbol{a}^T \tag{3.7}$$

Lastly, problem (3.7) is modified as:

$$\boldsymbol{Q}^* = \underset{\boldsymbol{Q}}{\operatorname{argmin}} - \log|\det\{\boldsymbol{Q}\}| + \lambda\|\boldsymbol{QY}\|_h, \quad \text{s.t.: } \mathbf{1}_p^T \boldsymbol{Q} = \boldsymbol{a}^T \tag{3.8}$$

where $\|\boldsymbol{X}\|_h = \sum_{ij} h\left([\boldsymbol{X}]_{ij}\right)$ and the *hinge function* $h(x) = \max\{-x, 0\}$. The *regularizer* term $\|\boldsymbol{QY}\|_h$ penalizes the negative components of $\boldsymbol{QY}$ proportionally to their mangitude controlled by *regularization parameter* $\lambda > 0$, and thus imposing a *soft constraint* instead of the hard non-negative constraint as before. This soft constraint yields solutions which are robust to outlier, noise and poor initialization, and can handle large scale problems.

Set an initial vector $\boldsymbol{q} = \texttt{vec}(\boldsymbol{Q})$ to denote the stack of columns in matrix $\boldsymbol{Q}$. Rewriting problem (3.8) in the following manner:

$$\boldsymbol{q}^* = \underset{\boldsymbol{q}}{\operatorname{argmin}}\, f(\boldsymbol{q}) + \lambda\|\boldsymbol{Aq}\|_h, \quad \text{s.t.: } \boldsymbol{Bq} = \boldsymbol{a} \tag{3.9}$$

where $f(\boldsymbol{q}) = -\log|\det\{\boldsymbol{Q}\}|$, $\boldsymbol{A} = \boldsymbol{Y}^T \otimes \boldsymbol{I}$ and $\boldsymbol{B} = \boldsymbol{I} \otimes \mathbf{1}_p^T$. The operator $A \otimes B$ denotes the Kronecker Tensor Product, implemented in the MATLAB function $\texttt{kron}(A, B)$. This problem is, however, non-convex. It is approximated by the first convex subproblem by computing a descent sequence $\boldsymbol{q}$ for $k = 0, 1, \ldots$.

Algorithm 8 SISAL - initialization and constraints [77]

Inputs:

$\boldsymbol{Y} \in \mathbb{R}^{p \times p}$;

K is the total number of iterations;

choose $\mu > 0$, i.e. augmented Lagrangian regularizer parameter;

choose $\lambda > 0$, i.e. Lagrangian multiplier;

Initialize:

$\boldsymbol{Q} \leftarrow \texttt{vca}(\boldsymbol{Y})$, good initialization (see algorithm 7);

$\boldsymbol{q_0} \leftarrow \texttt{vec}(Q)$, in MATLAB implemented as $q = Q(:)$;

$\boldsymbol{A} \leftarrow \texttt{kron}(\boldsymbol{Y}^T, \boldsymbol{I})$; $\quad \boldsymbol{B} \leftarrow \texttt{kron}(\boldsymbol{I}, \boldsymbol{1}_p^T)$;

for $k = 0$ to $K - 1$ **do**

$l_k \leftarrow f(\boldsymbol{q}_k) + \lambda \|\boldsymbol{A}\boldsymbol{q}_k\|_h$, evaluate objective function;

$\boldsymbol{g} \leftarrow -\texttt{vec}(\boldsymbol{Q}^{-1})$, the gradient of $f(\boldsymbol{q})$;

$\boldsymbol{q}_{k+1} = \text{argmin}_{\boldsymbol{q}}\, \boldsymbol{g}^T\boldsymbol{q} + \mu\|\boldsymbol{q} - \boldsymbol{q}_k\|_2^2 + \lambda\|\boldsymbol{A}\boldsymbol{q}\|_h$, s.t.: $\boldsymbol{B}\boldsymbol{q} = \boldsymbol{a}$, where $f(\boldsymbol{q})$ is replaced by quadratic approximation;

7: **if** $f(\boldsymbol{q}_{k+1}) + \lambda\|\boldsymbol{A}\boldsymbol{q}_{k+1}\|_h > l_k$ **then**

find $\boldsymbol{q}^* \leftarrow \{\alpha\boldsymbol{q}_{k+1} + (1-\alpha)\boldsymbol{q}_k : 0 < \alpha < 1\}$ s.t. $f(\boldsymbol{q}^*) + \lambda\|\boldsymbol{A}\boldsymbol{q}^*\|_h \leq l_k$;

$\boldsymbol{q}_{k+1} \leftarrow \boldsymbol{q}^*$

end if

end for

return $\boldsymbol{q}_K$

The quadratic term $\|\boldsymbol{q} - \boldsymbol{q}_k\|_2^2$ on the line computing $\boldsymbol{q}_{k+1}$, ensures that $\|\boldsymbol{q}_{k+1} - \boldsymbol{q}_k\|_2^2$ does not grow unbounded. The if statement starting on lines 7 ensures that the objective function does not increase. In the next subprogram the variable splitting augmented Lagrangians algorithm is introduced.

The optimization problem (3.9) of algorithm 8 is equivalent to the following formulation [77]:

$$(\boldsymbol{q}^*, \boldsymbol{z}^*) = \underset{\boldsymbol{q},\boldsymbol{z}}{\text{argmin}}\, E(\boldsymbol{q}, \boldsymbol{z}) \tag{3.10}$$

$$= \underset{\boldsymbol{q},\boldsymbol{z}}{\text{argmin}}\, \boldsymbol{g}^T\boldsymbol{q} + \mu\|\boldsymbol{q} - \boldsymbol{q}_k\|_2^2 + \lambda\|\boldsymbol{z}\|_h \tag{3.11}$$

$$\text{s.t.: } \boldsymbol{B}\boldsymbol{q} = \boldsymbol{a},\ \boldsymbol{A}\boldsymbol{q} = \boldsymbol{z} \tag{3.12}$$

The variable $\boldsymbol{q}$ is split into the pair $(\boldsymbol{q}, \boldsymbol{z})$ in (3.10) and are linked through the constraint $\boldsymbol{A}\boldsymbol{q} = \boldsymbol{z}$. Thus the *augmented Lagrangian* for this problem with respect to the constraint $\boldsymbol{A}\boldsymbol{q} = \boldsymbol{z}$ is as follows:

$$\mathcal{L}(\boldsymbol{q}, \boldsymbol{z}, \boldsymbol{d}, \tau) = E(\boldsymbol{q}, \boldsymbol{z}) + \boldsymbol{\alpha}^T(\boldsymbol{A}\boldsymbol{q} - \boldsymbol{z}) + \tau\|\boldsymbol{A}\boldsymbol{q} - \boldsymbol{z}\|_2^2 \tag{3.13}$$

$$= E(\boldsymbol{q}, \boldsymbol{z}) + \tau\|\boldsymbol{A}\boldsymbol{q} - \boldsymbol{z} - \boldsymbol{d}\|_2^2 + c \tag{3.14}$$

where vector α contains the Lagrange multipliers, vector $\boldsymbol{d} = -\boldsymbol{\alpha}/(2\tau)$, and c is an arbitrary constant. The augmented Lagrangian algorithm minimizes $\mathcal{L}$ with respect to $(\boldsymbol{q}, \boldsymbol{z})$, and updates $\boldsymbol{\alpha}$ and $\boldsymbol{d}$ simultaneously:

Algorithm 9 SISAL - Variable Splitting with Augmented Lagrangian [77]

Inputs:

$\boldsymbol{q}_0 \leftarrow \boldsymbol{q}$ returned by algorithm 8;

$\boldsymbol{z}_0 = \texttt{vec}(\boldsymbol{Q}\boldsymbol{Y})$, where $\boldsymbol{Q} \leftarrow \texttt{reshape}(\boldsymbol{q}, p, p)$;

$\boldsymbol{\alpha}_0 = \boldsymbol{0}_{p^2}$;

$\boldsymbol{\tau} > 0$, default $\boldsymbol{\tau} = 1$ for soft constraint;

K number of iterations;

Initialize:

$\boldsymbol{d}_0 \leftarrow \boldsymbol{\alpha}_0/(2\tau)$;

for $k = 0$ to $K - 1$ **do**

$(\boldsymbol{q}_{k+1}, \boldsymbol{z}_{k+1}) \leftarrow \text{argmin}_{\boldsymbol{q},\boldsymbol{z}}\, \mathcal{L}(\boldsymbol{q}, \boldsymbol{z}, \boldsymbol{d}_k, \tau)$, s.t.: $\boldsymbol{B}\boldsymbol{q} = \boldsymbol{a}$;

$\boldsymbol{d}_{k+1} \leftarrow \boldsymbol{d}_k - (\boldsymbol{A}\boldsymbol{q}_{k+1} - \boldsymbol{z}_{k+1})$;

end for

return $(\boldsymbol{q}_K, \boldsymbol{z}_K)$

Note that the $\texttt{reshape}(\boldsymbol{X}, r, c)$ refers to the MATLAB function which can reshape a column vector back into 2-D matrix and vice versa.

Computing the sequence $\boldsymbol{d}_k$ in algorithm 9 is complex. Therefore, Bioucas-Dias proposed the modification to algorithm 9 by alternating the optimization of the split variables, based on Eckstein and Bertsekas's theorem 1:

Algorithm 10 SISAL - Alternating Split with Augmented Lagrangian [77]

Inputs:

$\boldsymbol{q}_0 \leftarrow \boldsymbol{q}$ returned by algorithm 8;

$\boldsymbol{z}_0 = \texttt{vec}(\boldsymbol{QY})$, where $\boldsymbol{Q} \leftarrow \texttt{reshape}(\boldsymbol{q}, p, p)$;

$\boldsymbol{\alpha}_0 = \mathbf{0}_{p^2}$;

$\boldsymbol{\tau} > 0$, default $\boldsymbol{\tau} = 1$ for soft constraint;

K number of iterations;

Initialize:

$\boldsymbol{d}_0 \leftarrow \boldsymbol{\alpha}_0/(2\tau)$;

for $k = 0$ to $K - 1$ **do**

$\boldsymbol{q}_{k+1} \leftarrow \text{argmin}_{\boldsymbol{q}}\, \boldsymbol{g}^T\boldsymbol{q} + \frac{\mu}{2}\|\boldsymbol{q} - \boldsymbol{q}_k\|_2^2 + \frac{\tau}{2}\|\boldsymbol{Aq} - \boldsymbol{z}_k - \boldsymbol{d}_k\|_2^2$, s.t.: $\boldsymbol{Bq} = \boldsymbol{a}$;

$\boldsymbol{z}_{k+1} \leftarrow \text{argmin}_{\boldsymbol{z}}\, \frac{1}{2}\|\boldsymbol{Aq}_{k+1} - \boldsymbol{z} - \boldsymbol{d}_k\|_2^2 + \frac{\lambda}{\tau}\|\boldsymbol{z}\|_h$;

$\boldsymbol{d}_{k+1} \leftarrow \boldsymbol{d}_{k+1} - (\boldsymbol{Aq}_{k+1} - \boldsymbol{z}_{k+1})$;

end for

return $(\boldsymbol{q}_K, \boldsymbol{z}_K)$

The solution $\boldsymbol{q}_{k+1}$ to the quadratic problem with linear constraints in the modified algorithm 10 is computed by Bioucas-Dias as follows:

$$\boldsymbol{q}_{k+1} = \boldsymbol{F}^{-1}\boldsymbol{b} - \boldsymbol{F}^{-1}\boldsymbol{B}^T(\boldsymbol{BF}^{-1}\boldsymbol{B}^T)^{-1}(\boldsymbol{BF}^{-1}\boldsymbol{b} - \boldsymbol{a}) \tag{3.15}$$

where

$$\boldsymbol{F} = (\mu\boldsymbol{I} + \tau\boldsymbol{A}^T\boldsymbol{A}) \tag{3.16}$$

$$\boldsymbol{b} = \mu\boldsymbol{q}_k - \boldsymbol{g} + \tau\boldsymbol{A}^T(\boldsymbol{z}_k + \boldsymbol{d}_k) \tag{3.17}$$

The minimization with respect to $\boldsymbol{z}$ in algorithm 10 is basically solving the convex hinge function $\|\boldsymbol{z}\|_h$ which is achieved by applying a soft threshold function to the negative part of its argument:

$$\boldsymbol{z}_{k+1} = \texttt{soft}_{-}(\boldsymbol{Aq}_{k+1} - \boldsymbol{d}_k, \mu/\tau) \tag{3.18}$$

where the function $\texttt{soft}_{-}$ is a proximal operator defined in [77] as follows:

$$\texttt{soft}_{-}(\boldsymbol{x}, \beta) = (\max\{|\boldsymbol{x} + \beta/2| - \beta/2, 0\})(\boldsymbol{x}/|\boldsymbol{x}|) \tag{3.19}$$

For further details on proximal operators see Parikh and Boyds's paper [101]. Algorithm 10 is faster than 9 and converges. Thus SISAL uses algorithm 10 as the augmented Lagrangian step.

Lastly, we obtain the estimated mixing matrix $\boldsymbol{M}$ with the following steps:

$$\boldsymbol{Q} = \texttt{reshape}(\boldsymbol{q}_K, p, p) \tag{3.20}$$

$$\boldsymbol{M} = \boldsymbol{Q}^{-1} \tag{3.21}$$

$$\boldsymbol{M} = \boldsymbol{U}_p \boldsymbol{M} \tag{3.22}$$

3.2.3 Sparse Unmixing Algorithms

Sparsity-based hyperspectral unmixing algorithms estimate abundance maps in the presence of over-complete spectral dictionaries, [9]. Thus we bypass the endmember extraction process, and we use a given dictionary *a priori*. Note that these methods are highly dependent on the mutual coherence of the dictionary and the dictionary's spectra is rarely acquired with the same conditions in this field.

Sparse unmixing methods make use of the fact that there is a very small number of endmembers making up a pixel compared to the large number of endmembers in a spectral library, [9]. Thus the abundance vector to be estimated for each pixel has only a few non-zero elements and thus is sparse.

Greedy algorithms such as the OMP have been used to unmix sparse data. Since the developments in sparse unmixing pioneered by Iordache, Bioucas-Dias and Plaza, the application of l_1-minimization in BP and BPDN algorithms with modified ADMM have gained popularity, [114]. These gave rise to the SUnSAL algorithm and its variations presented next.

Greedy Algorithms

Problem $\left(P_0^+\right)$ described in section 2.4.5 can be solved by the classic *orthogonal matching pursuit* (OMP) algorithm. The following pseudo-code is provided by Iordache in his Ph.D. thesis [16], which is based directly on Bruckenstein, Donoho and Elad's algorithm [97]. This is a *greedy* algorithm commonly used in compressive sensing, where it adds a new index to the target support (see section 2.2.1) and updates the target abundance vector as it best fits the measurements, i.e. minimizes the residual $\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_i$.

Note that the OMP algorithm computes the abundance vector $\boldsymbol{x}$ for one pixel $\boldsymbol{y} \in \boldsymbol{Y}$ at a time. The iterations for each pixel terminate once the residual is below a pre-specified threshold T. In Stephen Becker's implementation [115], the sparsity estimate $\hat{k}$ is used to iterate through the OMP algorithm, instead of a threshold.

Algorithm 11 Orthogonal Matching Pursuit [16], implemented by Stephen Becker

Inputs:

$\boldsymbol{y} \in \mathbb{R}^B$;

$\boldsymbol{A} \in \mathbb{R}^{B \times m}$, given dictionary;

T, i.e. threshold for stopping iteration;

Initialize:

$i \leftarrow 0$;

$\boldsymbol{x}_0 \leftarrow \mathbf{0}_p$, initial solution;

$\boldsymbol{r}_0 \leftarrow \boldsymbol{y}$, initial residual;

$\boldsymbol{\Lambda}_0 \leftarrow \emptyset$, initial support matrix;

while $\|\boldsymbol{r}_i\|_2^2 \leq T$ **do**

$i \leftarrow i + 1$;

$index \leftarrow \text{argmin}_{1 \leq k \leq m} \|\boldsymbol{A}_k \boldsymbol{x}_{i-1} - \boldsymbol{r}_{i-1}\|_2^2$, where $\boldsymbol{A}_k$ is the kth column of dictionary $\boldsymbol{A}_k$;

$\boldsymbol{\Lambda}_i \leftarrow \boldsymbol{\Lambda}_{i-1} \cup \{index\}$, update support;

$\boldsymbol{x}_i \leftarrow \text{argmin}_{\boldsymbol{x}} \|\boldsymbol{A}_{\boldsymbol{\Lambda}_i} \boldsymbol{x} - \boldsymbol{y}\|_2^2$, update solution;

$\boldsymbol{r}_i \leftarrow \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_i$, update residual;

end while

return $\boldsymbol{x}_i$.

The OMP algorithm can recover a k-sparse solution in k iterations. This is possible because the OMP selects one column from dictionary $\boldsymbol{A}$ which is the most correlated with the residual $\boldsymbol{r}$, and adds it to the set of columns $\boldsymbol{\Lambda}$. As mentioned above, instead of using a threshold to terminate iterations, Becker actually replaces the while loop in pseudocode 11 with a for loop iterating k times in his implementation. Subsequently the solution $\boldsymbol{x}$ is updated solving the least squares problem using the constructed $\boldsymbol{A}_{\boldsymbol{\Lambda}}$, and calculating a new residual for the following iteration.

OMP is limited in that it depends on the independence of the dictionaries columns, which is not always the case. Thus the OMP might select the wrong columns because of the high mutual coherence of the columns, and the recovered sparsity will not be correct.

The next algorithm we consider is Needell and Tropp's *compressive sampling matching pursuit* (CoSaMP), [116], that overcomes the problem of OMP not always being able to recover sparsity k. It also uses hard thresholding by keeping the absolute abundance values, while forcing the very small fractional abundances to be 0. The CoSaMP requires the sparsity estimate p, and it is advised to overestimate the sparsity, [40].

Algorithm 12 Compressed Sampling Matching Pursuit [116], implemented by Stephen Becker

Inputs:

$\boldsymbol{y} \in \mathbb{R}^B$;

$\boldsymbol{A} \in \mathbb{R}^{B \times m}$, given dictionary;

s, the sparsity level, i.e. the stopping criterion;

Initialize:

$\boldsymbol{x}_0 \leftarrow \mathbf{0}_s$, initial solution;

$\boldsymbol{r}_0 \leftarrow \boldsymbol{y}$, initial residual;

for k=1 to $s-1$ **do**

$\boldsymbol{\nu} \leftarrow \boldsymbol{A}^T \boldsymbol{r}_{k-1}$, signal proxy;

$\Omega \leftarrow \mathtt{supp}(\boldsymbol{\nu}_{2s})$, identify large components, see section 2.2.1;

$\boldsymbol{\Lambda} \leftarrow \Omega \cup \mathtt{supp}(\boldsymbol{x}_{k-1})$, merge supports;

$\boldsymbol{b}_\Lambda \leftarrow \mathrm{argmin}_{\boldsymbol{x}}\{\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2, \mathtt{supp}(\boldsymbol{x}) \subset \Lambda\}$

$\boldsymbol{x}_k \leftarrow H_s(\boldsymbol{b}_\Lambda)$, where H_s is the hard thresholding operator that keeps the s largest absolute entries an sets the others to 0;

$\boldsymbol{r}_k \leftarrow \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_k$, update residual;

end for

return $\boldsymbol{x}_k$.

Alternating Direction Method of Multipliers

Bioucas-Dias's *alternating direction method of multipiers* (ADMM) algorithm [98] is based on Eckstein and Bertsekas's theorem of convergence 1, considering problem (2.85):

Algorithm 13 Alternating Direction Method of Multipliers [98]

Inputs:

$\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{G} \in \mathbb{R}^{p \times n}$;

Initialize:

$k \leftarrow 0$, choose $\mu > 0$, $\boldsymbol{u}_0$ and $\boldsymbol{d}_0$;

repeat

$\boldsymbol{x}_{k+1} \leftarrow \mathrm{argmin}_{\boldsymbol{x}}\, f_1(\boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{G}\boldsymbol{x} - \boldsymbol{u}_k - \boldsymbol{d}_k\|_2^2$, where function $f_1 : \mathbb{R}^n \to \mathbb{R}$;

$\boldsymbol{u}_{k+1} \leftarrow \mathrm{argmin}_{\boldsymbol{u}}\, f_2(\boldsymbol{u}) + \frac{\mu}{2}\|\boldsymbol{G}\boldsymbol{x}_{k+1} - \boldsymbol{u} - \boldsymbol{d}_k\|_2^2$, where function $f_2 : \mathbb{R}^p \to \mathbb{R}$;

$\boldsymbol{d}_{k+1} \leftarrow \boldsymbol{d}_k - (\boldsymbol{G}\boldsymbol{x}_{k+1} - \boldsymbol{u}_{k+1})$;

$k \leftarrow k + 1$;

until stopping criterion is satisfied.

Next, the ADMM algorithm is customised to fit the various problems that were presented in section 2.4.5, which are based on the *basis pursuit* (BP) algorithm and variations of ADMM.

3.2.4 ADMM CSR: The SUnSAL Algorithm

The general *constrained sparse regression* (CSR) optimization problem $\left(P_1^{\delta+}\right)$ in (2.83) written in the equivalent form (2.84), [98]:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 + \iota_{\mathbb{R}_+^n}(\boldsymbol{x}) \tag{3.23}$$

where ι_S is the indicator function of the set S (i.e. $\iota_S(\boldsymbol{x}) = 0$ if $\boldsymbol{x} \in S$, and $\iota_S(\boldsymbol{x}) = \infty$ if $\boldsymbol{x} \notin S$) which enforces the non-negativity constraint. The regularization term $\|\boldsymbol{x}\|_1$ penalizes non-sparse solutions. Next we apply the following translations to the ADMM algorithm 13:

$$f_1(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \tag{3.24}$$

$$f_2(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1 + \iota_{\mathbb{R}_+^n}(\boldsymbol{x}) \tag{3.25}$$

$$\boldsymbol{G} = \boldsymbol{I} \tag{3.26}$$

The first quadratic problem in the ADMM algorithm 13 has the following solution:

$$\boldsymbol{x}_{k+1} = \boldsymbol{B}^{-1}\boldsymbol{w} \tag{3.27}$$

where

$$\boldsymbol{B} = \boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{I} \tag{3.28}$$

$$\boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{y} + \mu(\boldsymbol{u}_k + \boldsymbol{d}_k) \tag{3.29}$$

The second quadratic problem in the ADMM is simplified to the following:

$$\boldsymbol{u}_{k+1} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{\nu}_k\|_2^2 + \frac{\lambda}{\mu}\|\boldsymbol{u}\|_1 + \iota_{\mathbb{R}_+^n}(\boldsymbol{u}) \tag{3.30}$$

where $\boldsymbol{\nu}_k = \boldsymbol{x}_{k+1} - \boldsymbol{d}_k$. If the term $\iota_{\mathbb{R}_+^n}(\boldsymbol{u})$ were absent, then the solution would be the soft threshold function as defined (3.19):

$$\boldsymbol{u}_{k+1} = \texttt{soft}_-(\boldsymbol{\nu}_k, \lambda/\mu) \tag{3.31}$$

The ANC term $\iota_{\mathbb{R}_+^n}(\boldsymbol{u})$ gives the following solution:

$$\boldsymbol{u}_{k+1} = \max\{0, \texttt{soft}_-(\boldsymbol{\nu}_k, \lambda/\mu)\} \tag{3.32}$$

where the maximum function is applied component-wise.

Thus the following algorithm presents the ADMM algorithm for the CSR problem The general *constrained sparse regression* (CSR) optimization problem $\left(P_1^{\delta+}\right)$, called the *sparse unmixing by variable splitting and augmented Lagrangian* (SUnSAL) algorithm by replacing the quadratic problems in the ADMM algorithm 13 with solutions (3.27) and (3.32):

Algorithm 14 SUnSAL [98]

Inputs:

$\boldsymbol{y} \in \mathbb{R}^B$, $\boldsymbol{A} \in \mathbb{R}^{B\times m}$, $\lambda \geq 0$;

Initialize:

$k \leftarrow 0$, choose $\mu > 0$, $\boldsymbol{u}_0$ and $\boldsymbol{d}_0$;

repeat

$\boldsymbol{w} \leftarrow \boldsymbol{A}^T\boldsymbol{y} + \mu(\boldsymbol{u}_k + \boldsymbol{d}_k)$;

$\boldsymbol{B} \leftarrow \boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{I}$;

$\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{B}^{-1}\boldsymbol{w}$;

$\boldsymbol{\nu}_k \leftarrow \boldsymbol{x}_{k+1} - \boldsymbol{d}_k$

$\boldsymbol{u}_{k+1} \leftarrow \max\{0, \texttt{soft}_-(\boldsymbol{\nu}_k, \lambda/\mu)\}$;

$\boldsymbol{d}_{k+1} \leftarrow \boldsymbol{d}_k - (\boldsymbol{x}_{k+1} - \boldsymbol{u}_{k+1})$;

$k \leftarrow k + 1$;

until stopping criterion is satisfied.

According to Rockafellar's definitions of convex analysis [117], the cost function (3.23) is proper, convex, lower semi-continuous, and coercive. The functions f_1 and f_2 in (3.24) and (3.25) are close, and $\boldsymbol{G} = \boldsymbol{I}$ is full rank, thus theorem 1 can be invoked to ensure the convergence of the SUnSAL algorithm.

CLS and FCLS Algorithms

In terms of least squares approximation, for the *constrained least squares* (CLS) problem:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \iota_{\mathbb{R}_+^n}(\boldsymbol{x}) \tag{3.33}$$

simply input $\lambda = 0$ in the SUnSAL algorithm 14.

For the *fully constrained least squares* (FCLS) problem:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \iota_{\{1\}}\left(\mathbf{1}^T\boldsymbol{x}\right) + \iota_{\mathbb{R}_+^n}(\boldsymbol{x}) \tag{3.34}$$

where the term $\iota_{\{1\}}\left(\mathbf{1}^T\boldsymbol{x}\right)$ enforces the ASC $\mathbf{1}^T\boldsymbol{x} = 1$.

In addition to setting $\lambda = 0$ for FCLS, we have to modify the f_1 and f_2 functions, and the step computing $\boldsymbol{x}_{k+1}$ which is linked to the ASC in the ADMM algorithm. We apply the following translation table:

$$f_1(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \iota_{\{1\}}\left(\mathbf{1}^T\boldsymbol{x}\right) \tag{3.35}$$

$$f_2(\boldsymbol{x}) = \iota_{\mathbb{R}_+^n}(\boldsymbol{x}) \tag{3.36}$$

$$\boldsymbol{G} = \boldsymbol{I} \tag{3.37}$$

Therefore the solution for the quadratic problem in ADMM with $\lambda = 0$, and function (3.35) and (3.36), is as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{B}^{-1}\boldsymbol{w} - \boldsymbol{C}(\mathbf{1}^T\boldsymbol{B}^{-1}\boldsymbol{w} - 1) \tag{3.38}$$

where

$$\boldsymbol{B} = \boldsymbol{A}^T\boldsymbol{A} + \mu\boldsymbol{I} \tag{3.39}$$

$$\boldsymbol{C} = \boldsymbol{B}^{-1}\mathbf{1}(\mathbf{1}^T\boldsymbol{B}^{-1}\mathbf{1})^{-1} \tag{3.40}$$

$$\boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{y} + \mu(\boldsymbol{u}_k + \boldsymbol{d}_k) \tag{3.41}$$

The ANC term $\iota_{\mathbb{R}_+^n}(\boldsymbol{u})$ with input $\lambda = 0$ gives the following simplified solution:

$$\boldsymbol{u}_{k+1} = \max\{0, \boldsymbol{\nu}_k\} \tag{3.42}$$

Therefore, taking into account the modifications above, (3.38) and (3.42), the following shows the FCLS version of the SUnSAL algorithm:

Algorithm 15 SUnSAL for FCLS [98]

Inputs:

$\boldsymbol{y} \in \mathbb{R}^B$, $\boldsymbol{A} \in \mathbb{R}^{B \times m}$;

Initialize:

$k \leftarrow 0$, choose $\mu > 0$, $\boldsymbol{u}_0$ and $\boldsymbol{d}_0$;

repeat

$\boldsymbol{w} \leftarrow \boldsymbol{A}^T\boldsymbol{y} + \mu(\boldsymbol{u}_k + \boldsymbol{d}_k)$;

$\boldsymbol{B} \leftarrow \boldsymbol{A}^T\boldsymbol{A} + \mu\boldsymbol{I}$;

$\boldsymbol{C} \leftarrow \boldsymbol{B}^{-1}\mathbf{1}(\mathbf{1}^T\boldsymbol{B}^{-1}\mathbf{1})^{-1}$;

$\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{B}^{-1}\boldsymbol{w} - \boldsymbol{C}(\mathbf{1}^T\boldsymbol{B}^{-1}\boldsymbol{w} - 1)$;

$\boldsymbol{\nu}_k \leftarrow \boldsymbol{x}_{k+1} - \boldsymbol{d}_k$

$\boldsymbol{u}_{k+1} \leftarrow \max\{0, \boldsymbol{\nu}_k\}$;

$\boldsymbol{d}_{k+1} \leftarrow \boldsymbol{d}_k - (\boldsymbol{x}_{k+1} - \boldsymbol{u}_{k+1})$;

$k \leftarrow k + 1$;

until stopping criterion is satisfied.

Constrained Basis Pursuit Denoising

The *constrained basis pursuit* (CBP) problem $\left(P_1^+\right)$ corresponds to the *constrained basis pursuit denoising* (CBPDN) problem $\left(P_1^{\delta+}\right)$ by setting $\delta = 0$. Thus solving the CBPDN problem automatically solves the CBP as well. The CBPDN problem $\left(P_1^{\delta+}\right)$ is equivalent to the following formulation:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{x}\|_1 + \iota_{B(\boldsymbol{y},\delta)}(\boldsymbol{A}\boldsymbol{x}) + \iota_{\mathbb{R}^n_+}(\boldsymbol{x}) \tag{3.43}$$

where the indicator function $\iota_{B(\boldsymbol{y},\delta)}(\boldsymbol{z}) = \{\boldsymbol{z} : \|\boldsymbol{z} - \boldsymbol{y}\|_2^2 \leq \delta\}$ and B is the ball closed around $\boldsymbol{y}$ with a radius δ. We use the following definitions when applying ADMM:

$$f_1(\boldsymbol{x}) = 0 \tag{3.44}$$

$$f_2(\boldsymbol{u}) = \iota_{B(\boldsymbol{y},\delta)}(\boldsymbol{u}_1) + \lambda\|\boldsymbol{u}_2\|_1 + \iota_{\mathbb{R}^n_+}(\boldsymbol{u}_2) \tag{3.45}$$

$$\boldsymbol{G} = [\boldsymbol{A}^T\boldsymbol{I}]^T \tag{3.46}$$

where $\boldsymbol{u} = [\boldsymbol{u}_1^T\boldsymbol{u}_2^T]^T$. With these definitions, the solution to the quadratic problem in ADMM becomes:

$$\boldsymbol{x}_{k+1} = \boldsymbol{B}^{-1}\boldsymbol{w} \tag{3.47}$$

where

$$\boldsymbol{B} = \boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{I} \tag{3.48}$$

$$\boldsymbol{w} = \boldsymbol{A}^T(\boldsymbol{u}_{1,k} + \boldsymbol{d}_{1,k}) + (\boldsymbol{u}_{2,k} + \boldsymbol{d}_{2,k}) \tag{3.49}$$

The split variables $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are decoupled and thus we have to solve two separate problems:

$$\boldsymbol{u}_{1,k} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{\nu}_{1,k}\|_2^2 + \iota_{B(\boldsymbol{y},\delta)}(\boldsymbol{u}) \tag{3.50}$$

$$\boldsymbol{u}_{2,k} = \underset{\boldsymbol{u}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{\nu}_{2,k}\|_2^2 + \frac{\lambda}{\mu}\|\boldsymbol{u}\|_1 + \iota_{\mathbb{R}_+^n}(\boldsymbol{u}) \tag{3.51}$$

where

$$\boldsymbol{\nu}_{1,k} = \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{d}_{1,k} \tag{3.52}$$

$$\boldsymbol{\nu}_{2,k} = \boldsymbol{x}_{k+1} - \boldsymbol{d}_{2,k} \tag{3.53}$$

The solution for (3.50) is obtained by projecting onto the ball $B(\boldsymbol{y}, \delta)$:

$$\boldsymbol{u}_{1,k+1} = \psi_B(\boldsymbol{y}, \delta)(\boldsymbol{\nu}_{1,k}) = \begin{cases} \boldsymbol{\nu}_{1,k}, & \|\boldsymbol{\nu}_{1,k} - \boldsymbol{y}\|_2^2 \leq \delta \\ \boldsymbol{y} + \frac{\boldsymbol{\nu}_{1,k} - \boldsymbol{y}}{\|\boldsymbol{\nu}_{1,k} - \boldsymbol{y}\|_2^2}\delta, & \|\boldsymbol{\nu}_{1,k} - \boldsymbol{y}\|_2^2 > \delta \end{cases} \tag{3.54}$$

The solution for the second decoupled variable is given by:

$$\boldsymbol{u}_{2,k+1} = \max\{\boldsymbol{0}, \texttt{soft}_-(\boldsymbol{\nu}_{2,k}, \lambda/\mu)\} \tag{3.55}$$

The solutions (3.47), (3.54) and (3.55) allow us to implement the constrained SUnSAL (C-SUnSAL) algorithm for the CBPDN problem, by replacing the lines concerned with the solutions for the quadratic problem in the ADMM algorithm. The CBP problem is solved by simply setting $\delta = 0$ in the C-SUnSAL.

Algorithm 16 C-SUnSAL[98]

Inputs:

$\boldsymbol{y} \in \mathbb{R}^B$, $\boldsymbol{A} \in \mathbb{R}^{B \times m}$;

Initialize:

$k \leftarrow 0$, choose $\mu > 0$, $\boldsymbol{u}_{1,0}$, $\boldsymbol{d}_{1,0}$, $\boldsymbol{u}_{2,0}$, $\boldsymbol{d}_{2,0}$, and $\delta \geq 0$;

repeat

$\boldsymbol{w} \leftarrow \boldsymbol{A}^T(\boldsymbol{u}_{1,k} + \boldsymbol{d}_{1,k}) + (\boldsymbol{u}_{2,k} + \boldsymbol{d}_{2,k})$;

$\boldsymbol{B} \leftarrow \boldsymbol{A}^T\boldsymbol{A} + \boldsymbol{I}$;

$\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{B}^{-1}\boldsymbol{w}$;

$\boldsymbol{\nu}_{1,k} \leftarrow \boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{d}_{1,k}$;

$\boldsymbol{u}_{1,k+1} \leftarrow \psi_B(\boldsymbol{y}, \delta)(\boldsymbol{\nu}_{1,k})$;

$\boldsymbol{\nu}_{2,k} \leftarrow \boldsymbol{x}_{k+1} - \boldsymbol{d}_{2,k}$;

$\boldsymbol{u}_{2,k+1} \leftarrow \max\{\boldsymbol{0}, \texttt{soft}_-(\boldsymbol{\nu}_{2,k}, \lambda/\mu)\}$;

$\boldsymbol{d}_{1,k+1} \leftarrow \boldsymbol{d}_{1,k} - (\boldsymbol{A}\boldsymbol{x}_{k+1} - \boldsymbol{u}_{1,k+1})$;

$\boldsymbol{d}_{2,k+1} \leftarrow \boldsymbol{d}_{2,k} - (\boldsymbol{x}_{k+1} - \boldsymbol{u}_{2,k+1})$;

$k \leftarrow k + 1$;

until stopping criterion is satisfied.

Bioucas-Dias's `sunsal`() function custom-built for MATLAB [118], takes a number of parameters which sets up code to compute either the CSR, CLS, FCLS, BPDN or CBPDN algorithms. In my experiments, I have used most of these settings.

3.3 Summary of Algorithms

In summary, we provided pseudo-code to the following algorithms used in our experiments:

- Our methodology of simulating artificial toy data and realistic toy data based on i.i.d. dictionaries and USGS libraries, and artificial images based on Cuprite and Terrain images.
- Dimension reduction techniques comprised of signal noise estimation, subspace identification, and affine projection.
- Geometric endmember extraction algorithms based on the pure pixels assumption, on the near-pure pixels assumption, and then minimum volume simplex analysis.
- Sparse unmixing algorithms of greedy nature, and more importantly, the basis pursuit algorithms based on the alternating Lagrangian method of multipliers.

Chapter 4

Experiment Results and Comparative Analysis

We return to the main research questions laid out in chapter 1 which were:

- Which endmember extraction algorithm identifies the active sparse endmembers the best? Is there a tradeoff in performance versus complexity?
- Do available spectral dictionaries perform better than image-extracted dictionaries in reconstructing the image? What is the optimal sparsity of signals which are recoverable using hyperspectral libraries versus image-extracted dictionaries? That is, for which sparsity k do obtain successful signal reconstruction.
- Which sparse unmixing algorithm achieves the most accurate reconstruction on artificial toy data, realistic toy data, and on artificial images generated from real images? What is the probability of success?

This chapter aims to answer these questions, in the order stated above. First, we describe spectral libraries used in artificial data simulations and the limitations. The performance discriminators are presented next. Next we answer the first research question by analysing which endmember extraction algorithm (EEA) performed the best. Then we compare spectral libraries versus image-extracted dictionaries, and thus which is best suited for the sparse unmixing problem. Lastly we measure the reconstruction accuracy for various data and how these data configurations affect the unmixing.

Note that all the simulations were run in MATLAB R2018b on a machine with Intel Core i7-6500U CPU at 2.50 GHz with 8 GB of RAM and 64-bit OS.

4.1 Dictionaries used in Simulated Data Experiments

The dictionaries used in these experiments were specified earlier in section 3.1 in a general form. In this section we provide the specifications relevant to our simulations. It is important to note, that we adopted Bioucas-Dias and Iordache's testing framework. This is based on the fact that even though the spectral library may have hundreds of spectral signatures, an observed scene might only contain roughly 20 endmembers, and each pixel will contain even less, say 4 or 5 endmembers. The analyst would choose an appropriate subset of this library, with some prior knowledge of the scene being analysed.

This framework randomly selects $2 \times k$ spectral signature subsets from a library that we load in MATLAB, given a predefined sparsity $k \in \{2, 3, \ldots, 20\}$, same as Bioucas-Dias selected double the sparsity k in his code. Thus the dictionary subset used in our experiments would be in space $\mathbb{R}^{B\times(2\times k)}$, where B is the number of spectral bands acquired. This subset enables testing scenarios to be carried out faster. Here are the dictionaries used for simulating data, and for sparse unmixing algorithms according to Bioucas-Dias's demonstrations at the following links: http://www.lx.it.pt/ bioucas/code.htm and http://www.lx.it.pt/ bioucas/publications.html .

The dictionary specifications for our experiments are as follows:

i $\boldsymbol{A}_1 \in \mathbb{R}^{224\times(2\times k)}$ made of i.i.d. zero-mean components with variance of one chosen from a Gaussian distribution, with the number of spectral bands pre-specified to $B = 224$;

ii $\boldsymbol{A}_2 \in \mathbb{R}^{224\times(2\times k)}$ made of i.i.d. components uniformly distributed in the interval $[0, 1]$, with the number of spectral bands pre-specified to $B = 224$;

iii $\boldsymbol{A}_3 \in \mathbb{R}^{224\times(2\times k)}$, i.e. randomly selected subset of $2 \times k$ spectral signatures from the USGS 1995 spectral library in $\mathbb{R}^{224\times 498}$;

iv $\boldsymbol{A}_4 \in \mathbb{R}^{224\times(2\times k)}$, i.e. randomly selected subset of $2 \times k$ spectral signatures from the pruned USGS 1995 spectral library in $\mathbb{R}^{224\times 342}$, where the angle between any two different columns is larger than 3° to avoid signatures in USGS 1995 library that some may correspond to very small variations of the same material;

v $\boldsymbol{A}_5 \in \mathbb{R}^{188\times k}$ is based on an artificial hyperspectral image of the mineral-rich Cuprite site in Nevada acquired by AVIRIS in 1995. Bioucas-Dias removed the low SNR bands from the initial 224, thus only keeping 188 bands, and cropped

the image to have 250 lines and 191 columns, see figure 4.1. For each simulated dictionary instance, first we extract k endmembers from the Bioucas-Dias's cropped and cleaned image using the N-FINDR algorithm, then we synthesise the artificial image composed of only k endmembers, yet spatially correlated;

vi $\boldsymbol{A}_6 \in \mathbb{R}^{166\times k}$ is based on an artificial Terrain hyperspectral image acquired by the HYDICE sensor in 1993. Low SNR bands were also removed from the original 210 bands, thus leaving the image to contain only 166 band, see figure 4.2. The size of the terrain image has 500 lines and 307 columns. Each artificial dictionary is made up of k endmembers obtained by the same method as for $\boldsymbol{A}_5$.

where $k \in \{2, 3, \ldots, 20\}$ for all of the above library subsets. Note that for $\boldsymbol{A}_5$ and $\boldsymbol{A}_6$ we do not have to simulate image $\boldsymbol{Y}$ at every simulation, as it has already been prepared in the step creating the artificial image.

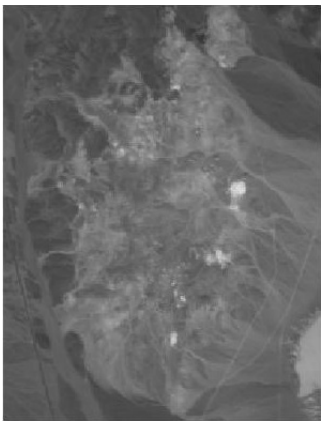


Fig. 4.1: Cuprite image acquired by AVIRIS, 1993.

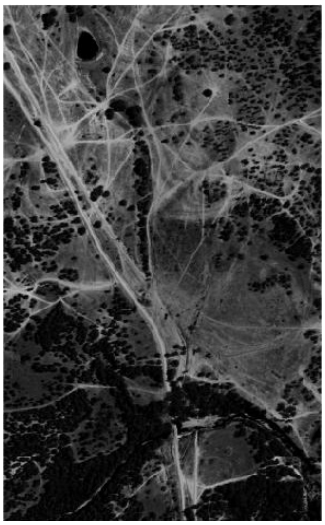


Fig. 4.2: Terrain image acquired by HYDICE, 1995

Libraries $\boldsymbol{A}_5$ and $\boldsymbol{A}_6$ contain the k extracted members via N-FINDR from the Cuprite and Terrain images. These are the only libraries where endmembers are not selected randomly from a distribution or from the available USGS library. These mixing matrices are based solely on real images.

Libray	**Description**	**Number of Bands** B	**Number of Spectra** m	$\texttt{spark}(\boldsymbol{A})$ Upper bound	$\mu(\boldsymbol{A})$
$\boldsymbol{A}_1$	i.i.d. Gaussian	224	440	223	0.27012
$\boldsymbol{A}_2$	i.i.d. Uniform	224	440	222	0.82897
$\boldsymbol{A}_3$	USGS	224	498	15	0.99998
$\boldsymbol{A}_4$	USGS pruned	224	342	16	0.99861
$\boldsymbol{A}_5$	Cuprite Image	188	20	13	0.99989
$\boldsymbol{A}_6$	Terrain Image	166	20	63	0.99951

Tab. 4.1: Mutual Coherence and Spark Estimates for Different Libraries

Table 4.1 shows a quick analysis of the properties of each dictionary used in our simulations. This analysis is put in context with the help of definitions and the accompanying graphs to follow below. Note that since $\boldsymbol{A}_5$ and $\boldsymbol{A}_6$ are not proper libraries, we extracted 20 endmembers for this example to represent the "spectral libraries" for the Cuprite and Terrain images.

The $\texttt{spark}(\boldsymbol{A})$ of a matrix was defined by Donoho and Elad [119] as the smallest number of linearly *dependent* columns of $\boldsymbol{A}$, i.e.:

$$\texttt{spark}(\boldsymbol{A}) = \min_{\boldsymbol{d}\neq\boldsymbol{0}} \|\boldsymbol{d}\|_0 \quad s.t.\ \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0} \tag{4.1}$$

In [9], Iordache explains that for noiseless data, the linear system $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ is expected to have a unique solution with non-zero entires $\|\boldsymbol{x}\|_0 < \texttt{spark}(\boldsymbol{A})/2$. For independent and identically distributed dictionaries we expect the spark to be very close to the number of spectral bands B, as the energy is spread fairly evenly over all the wavelengths, on average. Iordache states that for i.i.d. libraries, the linear system should have a unique solution with no more than $B/2$ non-zero entires with a probability of 1.

Computing a solution for problem (4.1) is a hard problem because of the l_0-norm. However, Iordache estimates the *upper bound* for the *spark* of a dictionary as the number of *discrete cosine transform* (DCT) coefficients whose accumulated energy explains 99.9% of the dictionary's energy, [9].

For the independent and identically distributed dictionaries $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, which have been used extensively in literature to validate the algorithms described in chapter 3, we observe in the cumulative energy plots in the left-side figures 4.3 and 4.4 that the cumulative energy tends to 99.9% as their spark upper bound gets closer

to B. The linear nature of the cumulative energy plot for $\boldsymbol{A}_2$ is due to the energy being uniformly distributed. The curvature of the cumulative energy plot for $\boldsymbol{A}_1$ is due to the Gaussian distribution of the energy.

Iordache computed that 99.9% of the energy is contained in the first 21 coefficients for the USGS library and 23 coefficients for the pruned USGS dictionary. We obtained slightly lower estimates for the spark upper bound than Iordache as shown in figures 4.5 and 4.6, however the point is that we also obtain spark estimates less than 20 DCT coefficients, that is substantially less than the number of spectral bands B available. This shows that the dictionary is sparse and can be represented with a much small number of DCT components than the number of bands. The Cuprite site has been found to contain 25 different minerals, however our image is cropped and has less materials as indicated by the spark for $\boldsymbol{A}_5$, see 4.7. As for the artificial Terrain library, the spark in figure 4.8 is rather high, but this could be due to noisy sensors.

We discuss the *mutual coherence* measure, which is the maximum absolute value of the cosine of the angle between any two columns of $\boldsymbol{A}$ given by:

$$\mu(\boldsymbol{A}) = \max_{1 \leq k,j \leq m,\, k \neq j} \frac{\left|\boldsymbol{a}_k^T \boldsymbol{a}_j\right|}{\|\boldsymbol{a}_k\|_2 \|\boldsymbol{a}_j\|_2} \tag{4.2}$$

The mutual coherence of libraries $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ is less, since the i.i.d. components are sampled randomly, and have no physical significance. It is important to note that libraries $\boldsymbol{A}_3$ to $\boldsymbol{A}_6$ have very high mutual coherence values close to 1, which is the very reason why typical blind source separation algorithms such as the ICA do not work on hyperspectral libraries. Table 4.1 tells us that mutual coherence for dictionary A_4 is slightly lower than A_3, since it has been pruned by 3°.

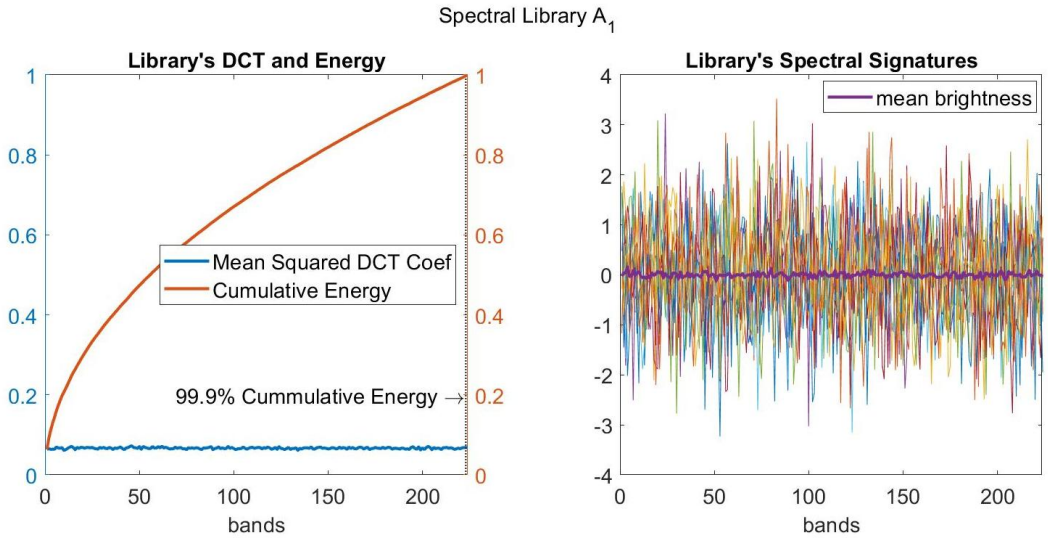


Fig. 4.3: (left) DCT coefficients and cummulative energy describing dictionary A_1; (right) A_1's mean signature and a few other randomly chosen signatures.

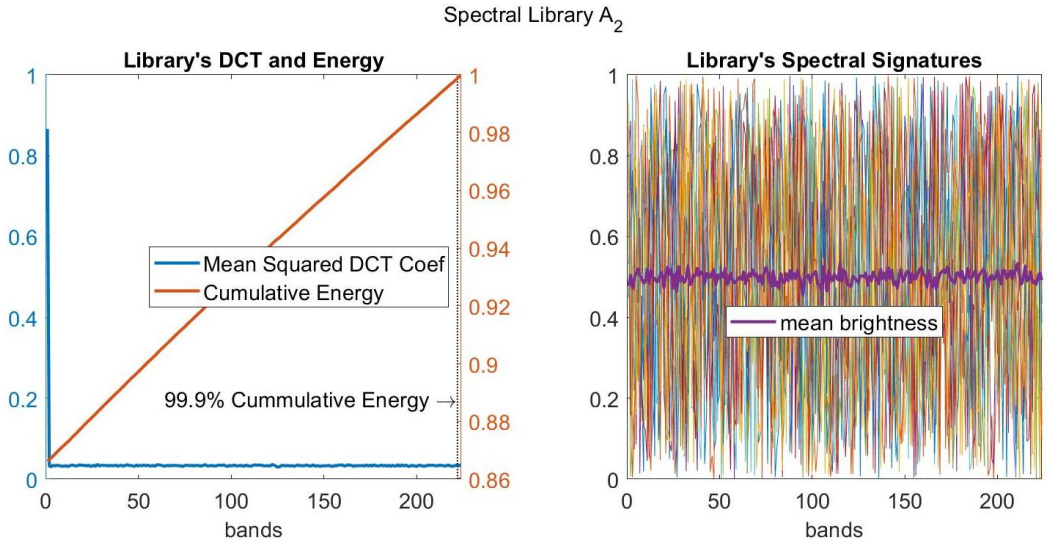


Fig. 4.4: (left) DCT coefficients and cummulative energy describing dictionary A_2; (right) A_2's mean signature and a few other randomly chosen signatures.

Even though the number of DCT coefficients is much smaller than the number of spectral bands, this estimate for the spark is an upper bound which is still higher than the expected low intrinsic dimension k. This suggests difficulties in the sparse

regression problem, which are mitigated by the very low level of sparsity of the signal $\boldsymbol{Y}$ that we are interested in unmixing, [9].

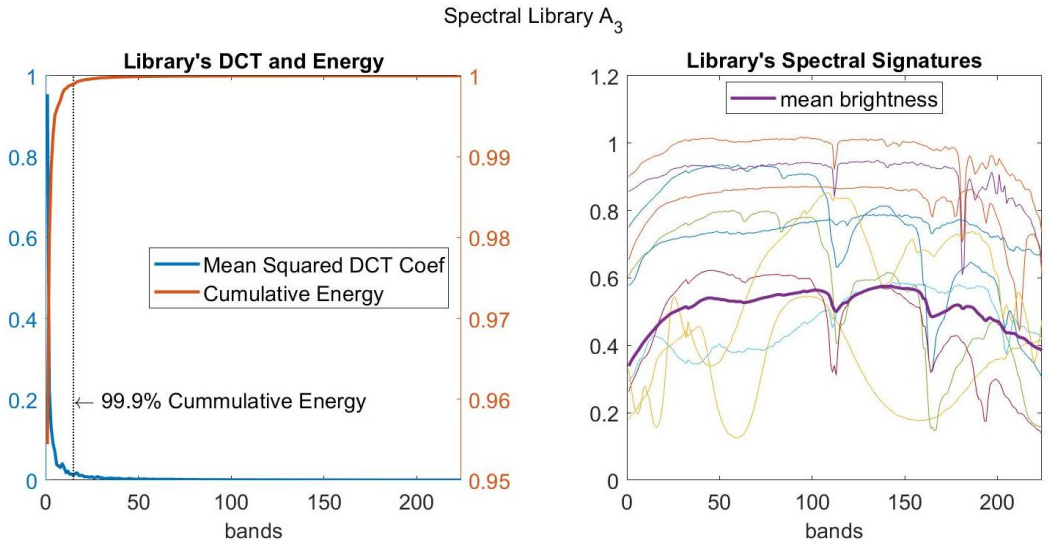


Fig. 4.5: (left) DCT coefficients and cummulative energy describing dictionary A_3; (right) A_3's mean signature and a few other randomly chosen signatures.

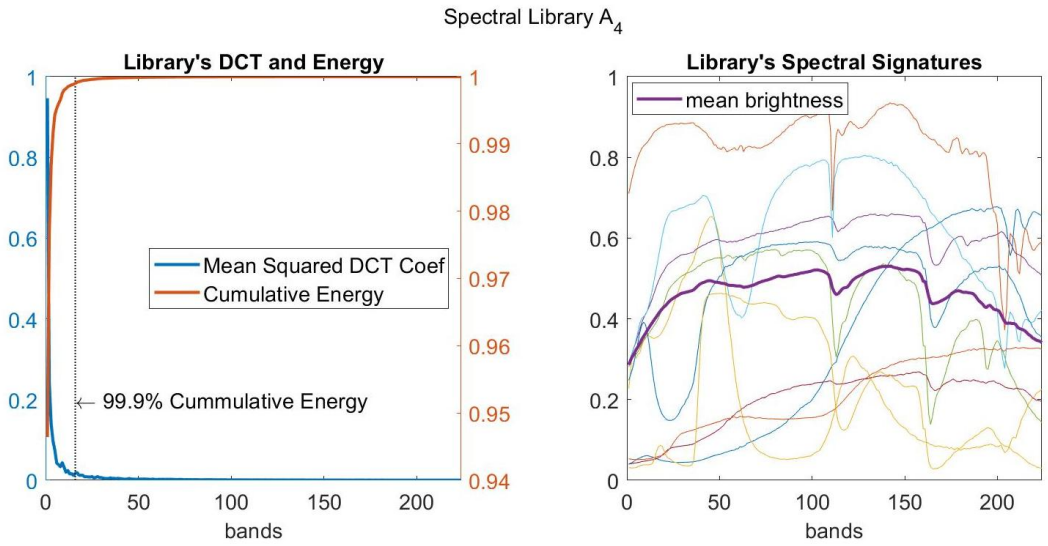


Fig. 4.6: (left) DCT coefficients and cummulative energy describing dictionary A_4; (right) A_4's mean signature and a few other randomly chosen signatures.

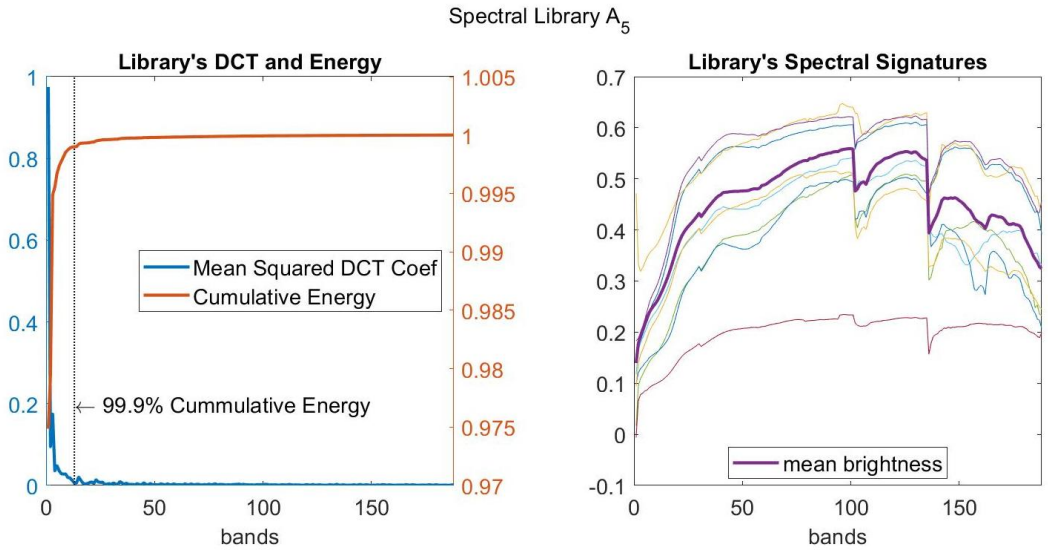


Fig. 4.7: (left) DCT coefficients and cummulative energy describing dictionary A_5; (right) A_5's mean signature and a few other randomly chosen signatures.

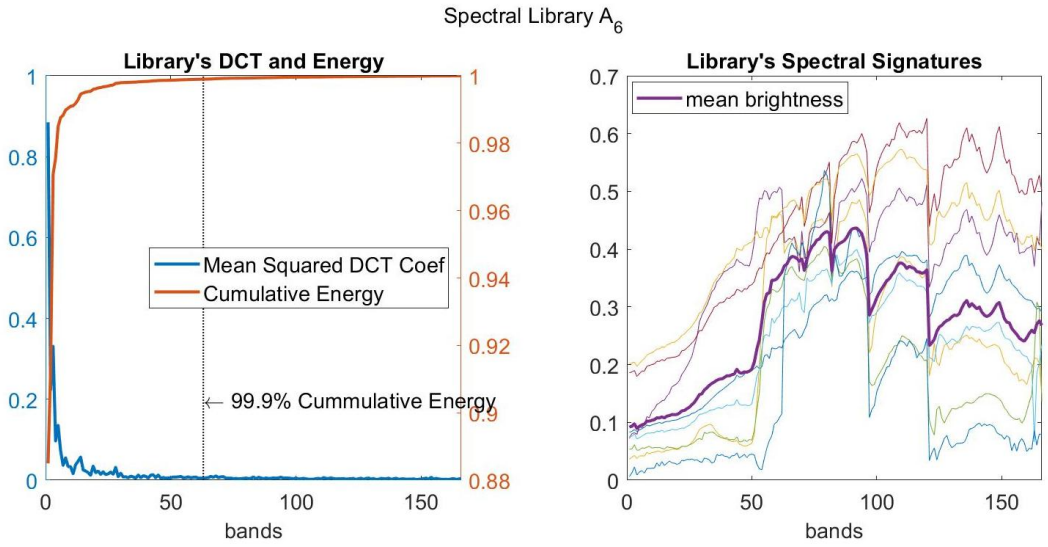


Fig. 4.8: (left) DCT coefficients and cummulative energy describing dictionary A_6; (right) A_6's mean signature and a few other randomly chosen signatures.

The right images in figures 4.7 and 4.8 show clearly that the spectral signatures for AVIRIS's Cuprite and HYDICE's Terrain images have similar envelopes, and thus explains the high mutual coherence. Also dictionary A_5 reaches 99.9% cumu-

lative energy in less that 20 DCT components, while A_6 takes just over 60 DCT components possibly due to noisy measurements.

4.2 Performance of Endmember Extraction Algorithms

We ran numerous scenarios with various parameters such as sparsity $k \in \{2, 3, \ldots, 20\}$, signal-to-noise ratio $SNR \in \{20, 30, 40, 50\}$ dB and libraries A_1 to A_6. The parameter $toy \in \{\text{true}, \text{false}\}$ refers to our choice of fractional abundances, indicating artificially simulated data or realistic simulated data configurations. The fractional abundances were randomly generated following a Dirichlet distribution, as explained in algorithm 1.

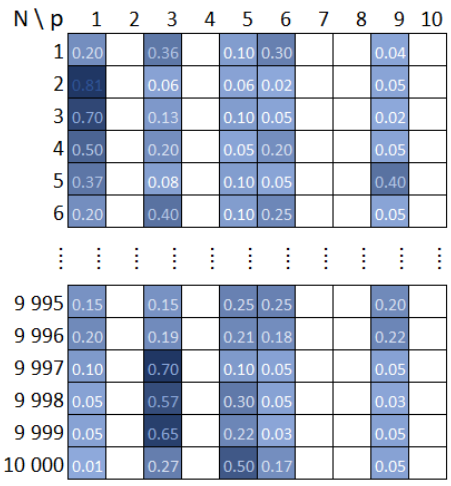


Fig. 4.9: Artificial fractional abundances with sparsity $k = 5$.

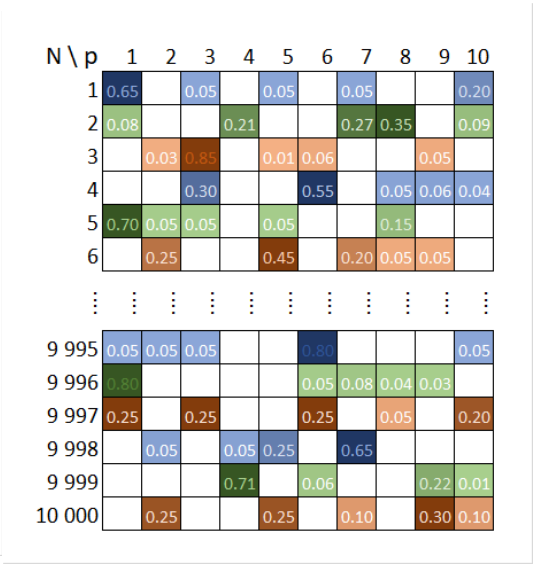


Fig. 4.10: Realistic fractional abundances with sparsity $k = 5$.

For a better visual explanation of how the fractional aboundances are generate, we have the following examples. Bioucas-Dias and Iordache's method is represented in figure 4.9 for generating artificial abundances for say $k = 5$. This method picks the same 5 endmembers throught out the simulation process. Our methods for genereating "realistic" but not spatially correlated fractional abundances is presented in figure 4.10. This figure should clarify that for "realistic" abundances, we

select 5 random endmembers at every iteration, i.e. for each pixel out of N pixels.

The random seed for MATLAB was set at the arbitrary value 123450, in order to replicate simulations exactly. The number of pixels for each simulated hyperspectral image was set to $N = 10,000$, except for Cuprite where $N = 47,750$, and $N = 153,500$ for the Terrain image.

4.2.1 Artificially Simulated Toy Data for various SNRs

For example, we consider the artificially simulated "toy" data versus realistically simulated "toy" data, with spectral signatures sampled from the USGS library $\boldsymbol{A}_3$ and the intrinsic dimension $k = 6$ per pixel. We compare the extracted endmembers under the worst case $SNR = 20\,dB$ versus the best case $SNR = 50\,dB$.

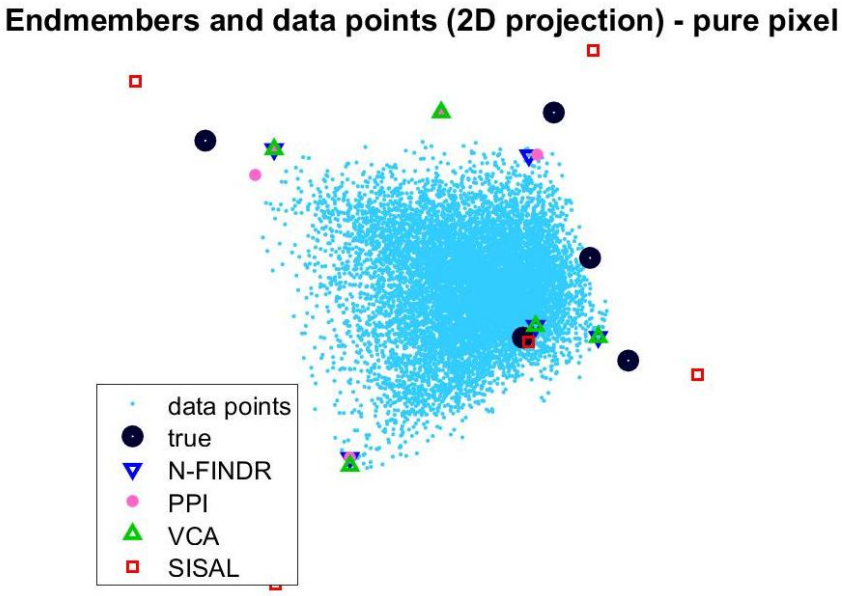


Fig. 4.11: 2D Projection of Extracted Endmembers and Artificial Toy Data, $SNR = 20\,dB$.

Due to the noisy data with $SNR = 20\,dB$, figure 4.11 shows the 2D projections of the extracted endmembers by the various EEAs far off the true endmembers. Even though the intrinsic dimension is set to $k = 6$ in the simulated data, the HySime algorithm 3 underestimated $\hat{k} = 5$. The EEAs only try to match 5 spectral signatures, as shown in figure 4.12 due to noisy data.

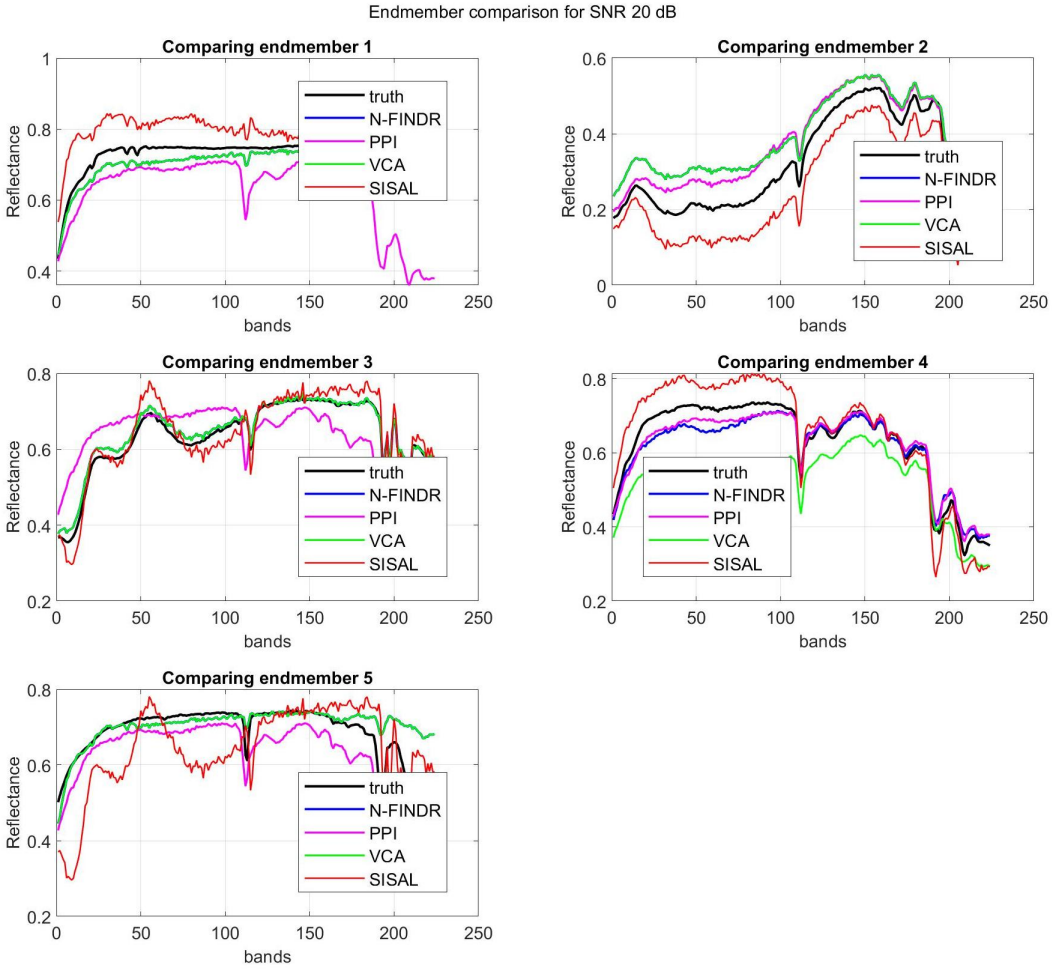


Fig. 4.12: Endmembers Extracted by Various EEAs for Artificial Toy Data with $SNR = 20\,dB$.

Comparing this to extracted endmembers for the "cleaner" simulated toy data with $SNR = 50\,dB$, the HySime algorithm estimates the correct sparsity of $\hat{k} = 6$, as we can see the SISAL extracted endmemeber are almost spot on the 2D projections of the true endmembers in figure 4.13. The SISAL algorithm produces the spectral signatures closest to the true signatures for this particular example as shown in figure 4.14.

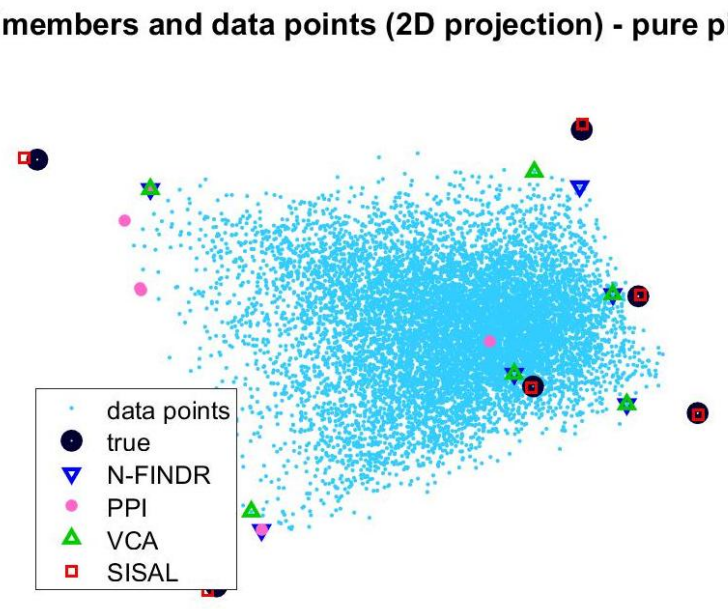


Fig. 4.13: 2D Projection of Extracted Endmembers and Artificial Toy Data, $SNR = 50\,dB$.

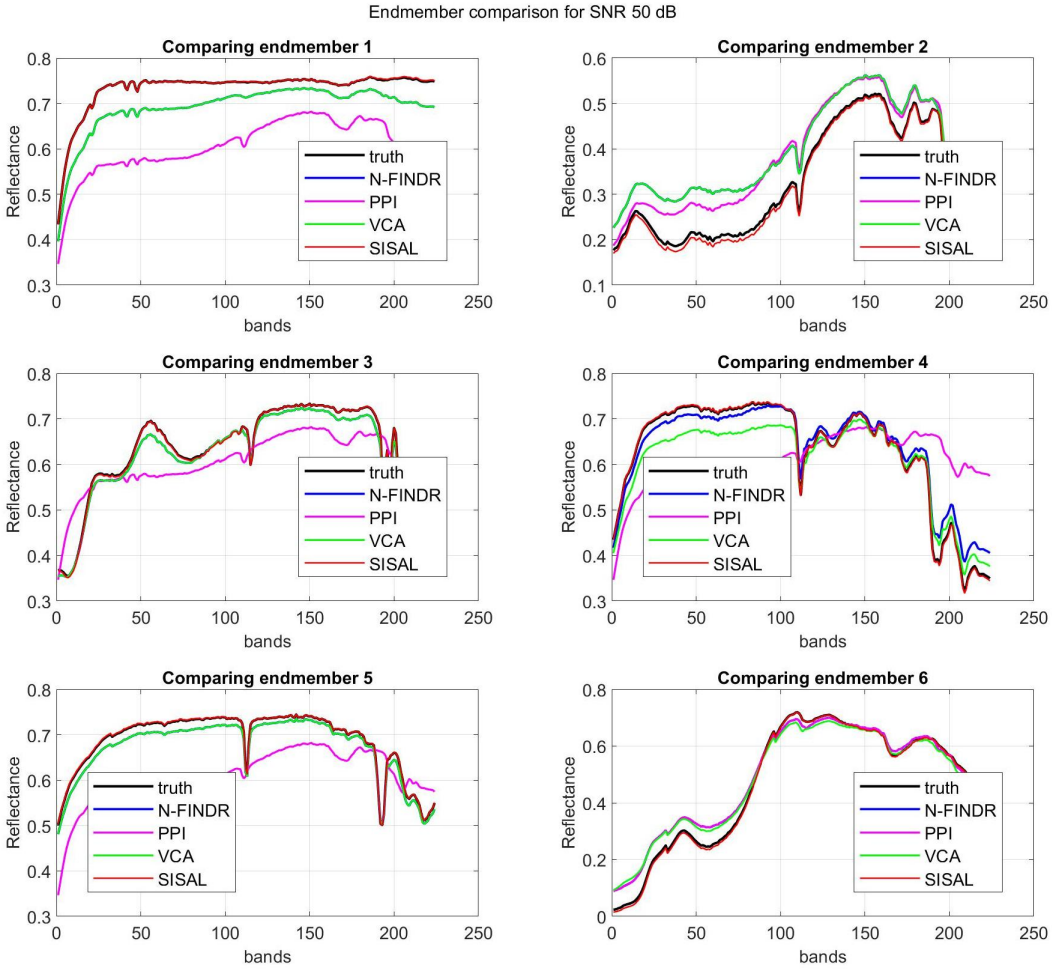


Fig. 4.14: Endmembers Extracted by Various EEAs for Artificial Toy Data with $SNR = 50\,dB$.

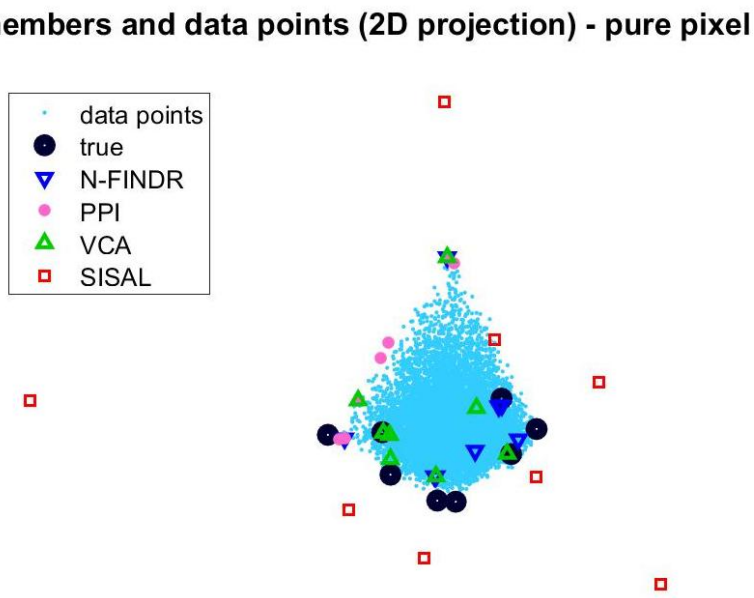


Fig. 4.15: 2D Projection of Extracted Endmembers and Realistic Toy Data, $SNR = 20\,dB$.

4.2.2 Realistically Simulated Toy Data for various SNRs

For realistically simulated toy data, where even though the sparsity is set to $k = 6$ *for each* pixel, the HySime algorithm estimates this intrinsic dimension closer to $2 \times k = 12$ as it decomposes the covariance matrix for the *whole* scene, not for individual pixels. Figure 4.15 shows that EEAs don't extract endmembers from noisy data close enough to the 2D projections of the true endmembers, while 4.17 gets closer to the 2D projections of the true endmembers of the *whole* scene.

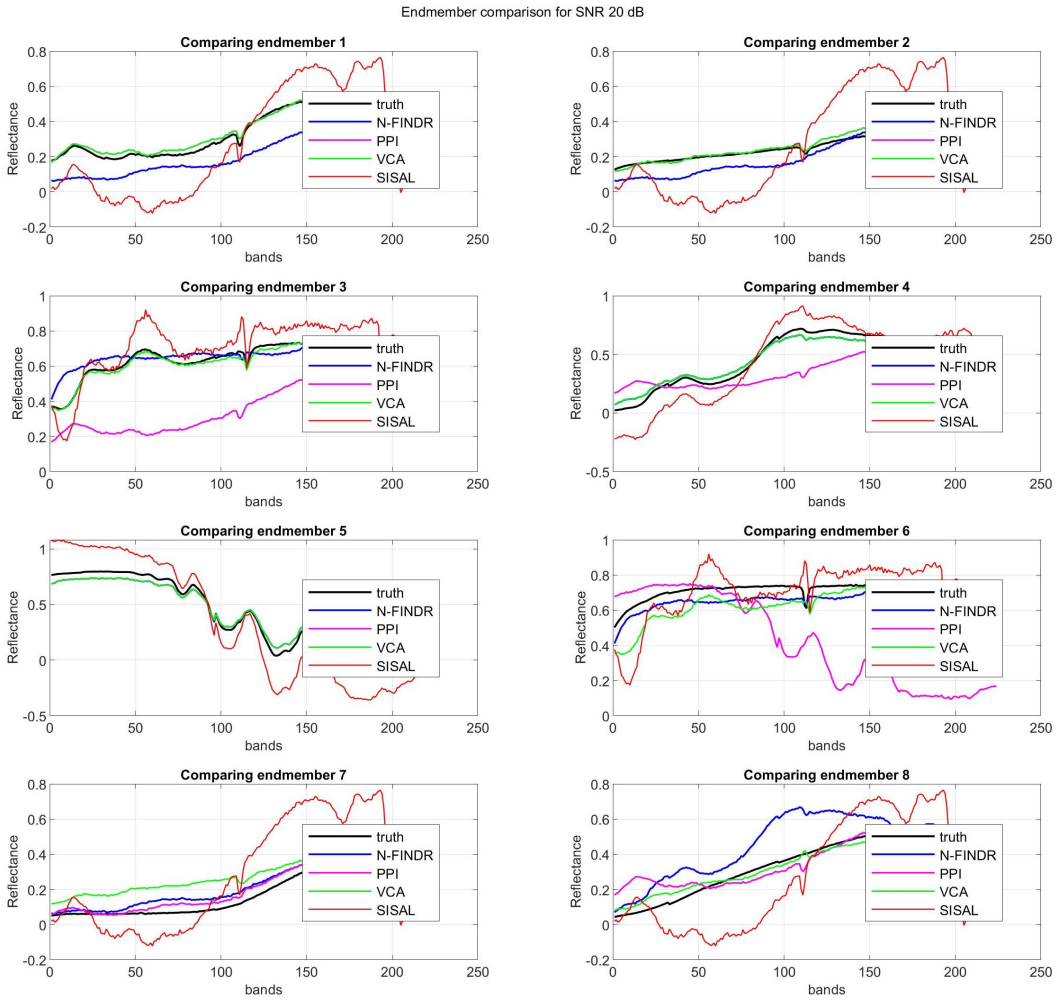


Fig. 4.16: Endmembers Extracted by Various EEAs for Realistic Toy Data with $SNR = 20\,dB$.

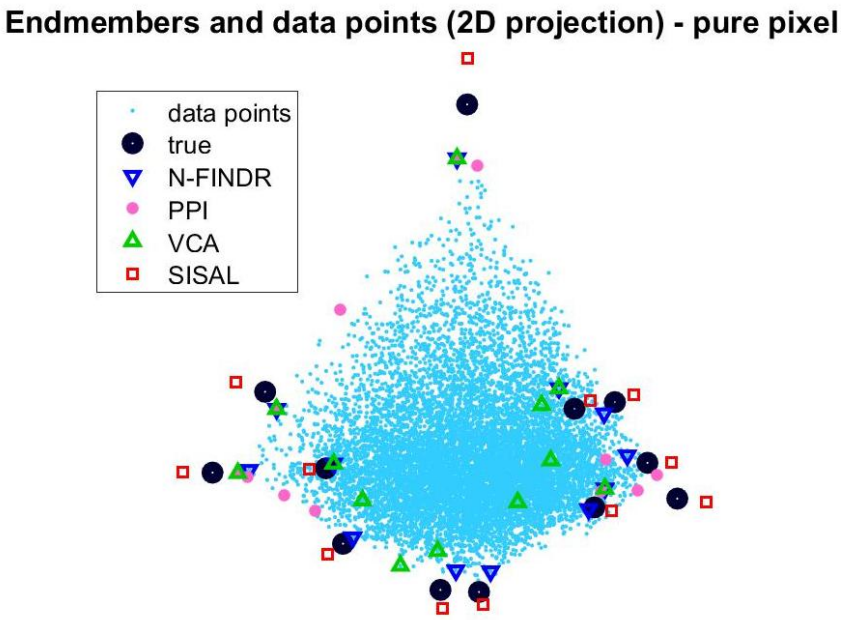


Fig. 4.17: 2D Projection of Extracted Endmembers and Realistic Toy Data, $SNR = 50\,dB$.

Therefore the signal-to-noise levels do impact the estimation of the intrinsic dimension $\hat{k}$, as for $SNR = 20\,dB$ the HySime algorithm estimates $\hat{k} = 8$, and for $SNR = 50\,dB$ it estimates $\hat{k} = 12$. Most notably in figures 4.16 and 4.20, the various signal-to-noise ratios impact which EEA peforms better. In this particular scenario for $k = 6$ and dictionary $\boldsymbol{A}_3$, for noisy $SNR = 20\,dB$, the SISAL algorithm extracts endmembers closest to the true signatures, while the VCA performs better for "cleaner" data.

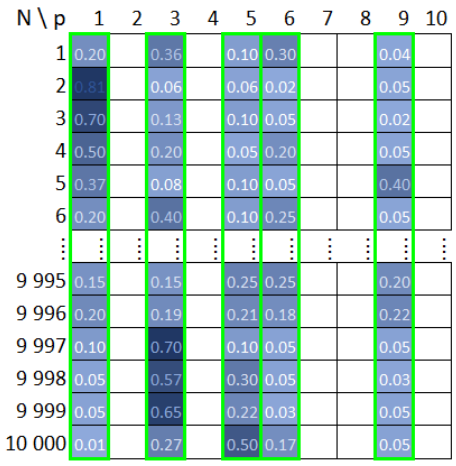


Fig. 4.18: Artificial fractional abundances with sparsity $k = 5$, instrinsic estimate of $\hat{k} = 5$.

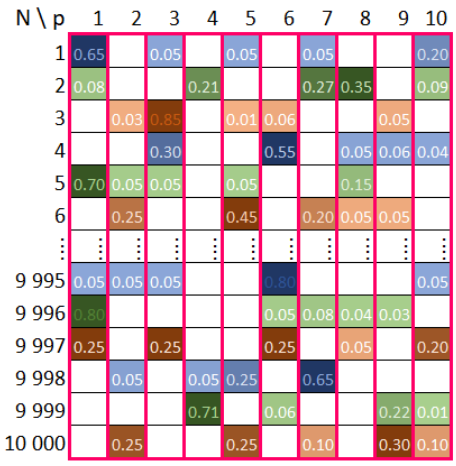


Fig. 4.19: Realistic fractional abundances with sparsity $k = 5$, instrinsic estimate of $\hat{k} = 10$.

The estimated intrinsic dimension $\hat{k} \to 2k$ depends on the Dirichlet distribution, and how the fractional abundances were simulated, as it can select any k signatures for each of the N pixels from library $\boldsymbol{A}_3$, which is made of $2k$ signatures.

Using the visual representation from before, we can clearly see that for simulated data having artificially simulated fractional abundances as in figure 4.18 the intrinsic estimate should be spot on. In comparison, for "realistically" generated fractional abundances as in figure 4.19 the intrisic dimension will be almost double the initial sparsity k that each pixel was generated with.

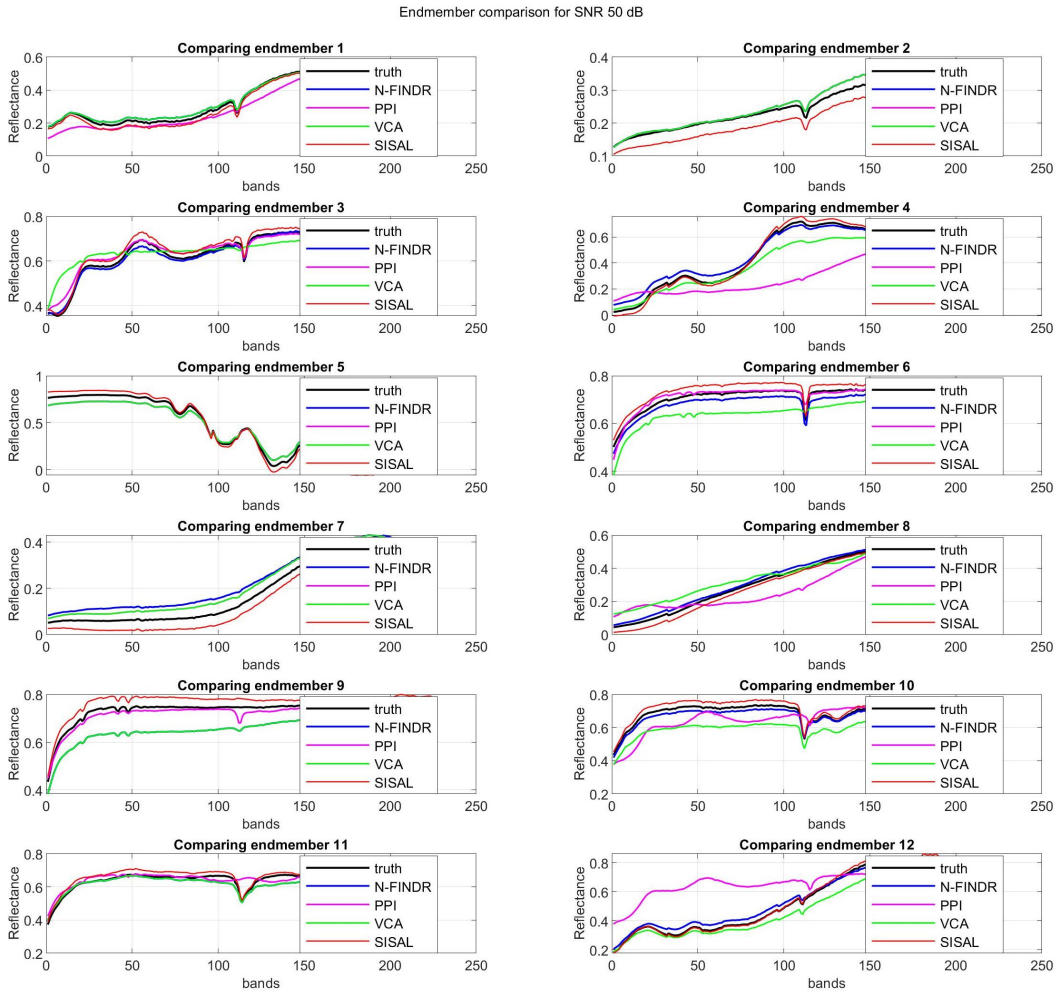


Fig. 4.20: Endmembers Extracted by Various EEAs for Realistic Toy Data with $SNR = 50\,dB$.

However, we cannot generalise from just one scenario where a particular EEA performs best. We need to obtain the all encompassing picture for all the scenarios that we considered to decide which are the better EEA and under which circumstances. We used the estimates of the intrinsic dimension $\hat{k}$ and the RMSE measure to decide the EEAs fate.

Thus we summarise next the matches and misses for HySime's performance of estimating the sparsity $\hat{k}$. For the i.i.d. libraries $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, the HySime algorithm gave us perfect matches for artificial toy data. The algorithm returned perfect $2k$ matches for the realistic toy data due to manner in which the fractional abundances are distributed.

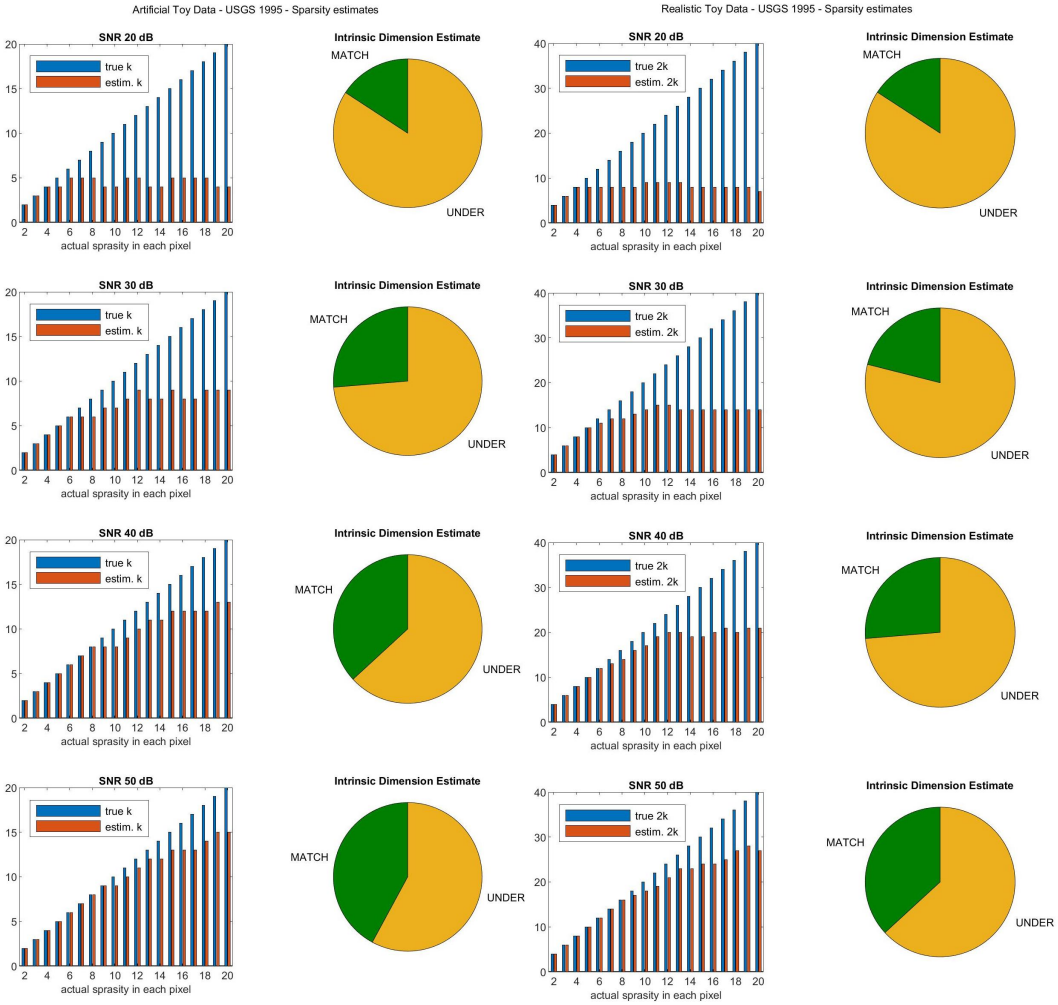


Fig. 4.21: HySime's mainly underestimates for artificial toy data from library $\boldsymbol{A}_3$.

Fig. 4.22: HySime's estimates for realistic toy data from library $\boldsymbol{A}_3$ tends closer to $2k$.

For artificial toy data simulated from library $\boldsymbol{A}_3$, HySime's intrinsic dimension estimates presented in figure 4.21 show mainly underestimated $\hat{k}$ due to the high mutual coherence of the USGS spectral signatures, thus the algorithm confuses some of the spectral signatures. A lot more matches were obtained for the pruned library

$\boldsymbol{A}_4$ in figure 4.23, since there is an imposed 3° angular difference between the spectral signatures, and thus not as mutually coherent as for the full USGS library. In general, the less noisy the artificial toy data, the better the estimate of the sparsity $\hat{k}$.

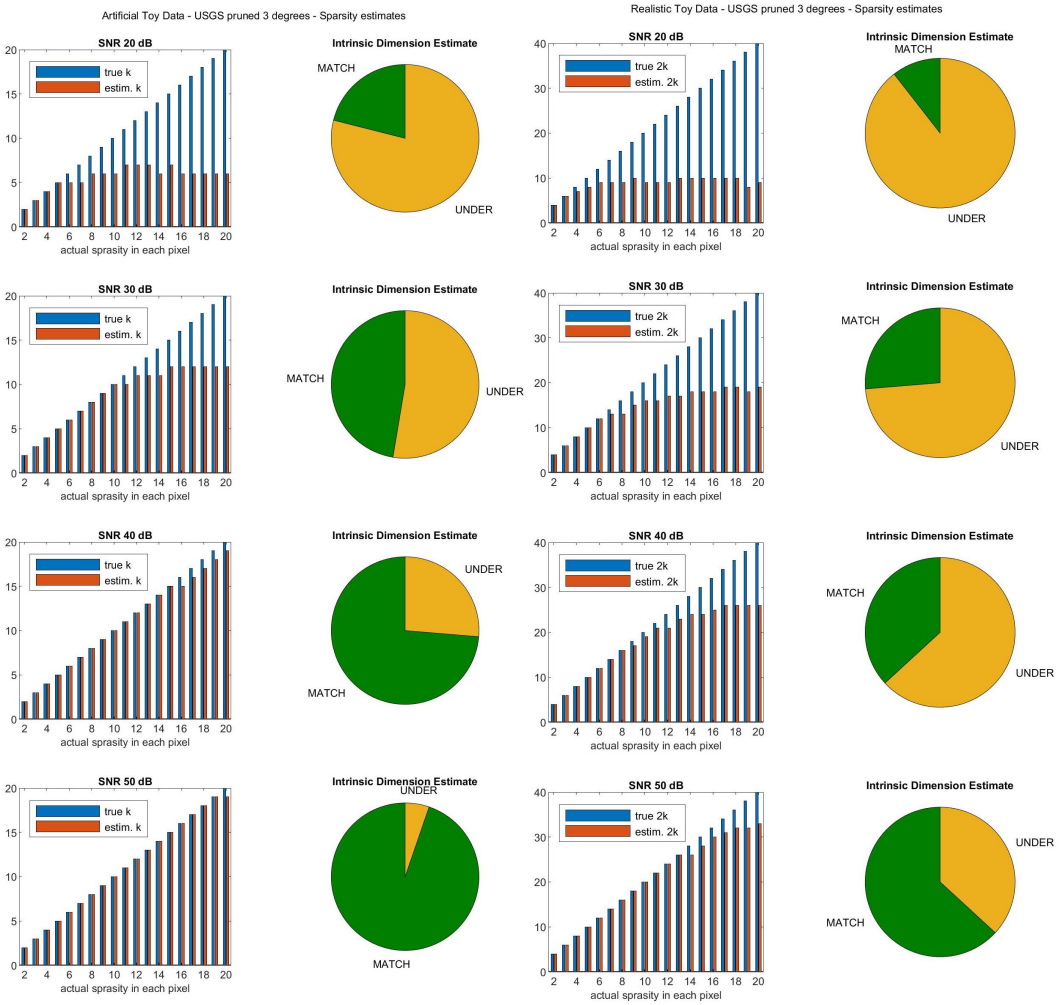


Fig. 4.23: HySime's mainly underestimates for artificial toy data from library $\boldsymbol{A}_4$.

Fig. 4.24: HySime's mainly underestimates for realistic toy data from library $\boldsymbol{A}_4$.

HySime's performance on noisier realistic toy data simulated from libraries $\boldsymbol{A}_3$ in figure 4.22 and $\boldsymbol{A}_4$ in figure 4.24 mostly underestimated the intrinsic dimension $2k$ as expected, and for less noisy data $\hat{k} \to 2 \times k$, especially for data based on $\boldsymbol{A}_4$.

4.2.3 Artificial Cuprite and Terrain Images

For the Cuprite and Terrain images we did not scramble the endmembers as we did for the realistic toy images, thus having the same arrangement of k endmembers for every pixel as in the artificial toy images. The artificial images also kept their inherent SNR, thus only one set of results for each image in figures 4.25 and 4.26.

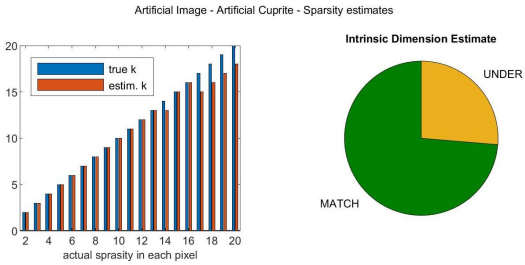


Fig. 4.25: HySime's very good matches for the Cuprite image $\boldsymbol{A}_5$.

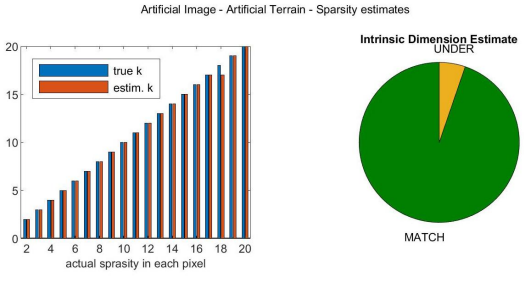


Fig. 4.26: HySime's almost perfect matches for the Terrain image $\boldsymbol{A}_6$.

The performance discriminator that we chose to measure how well the EEAs extracted the spectral signatures, compared to the original spectral signatures we used in simulating the data, was the *root-mean-squared error* (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{a}_i - a_i)^2} \tag{4.3}$$

where $\hat{a}_i$ is the row element of a column of estimated $\hat{\boldsymbol{a}}$ compared against the true endmember $\boldsymbol{a}$. Thus we search for the EEA giving the minimum RMSE.

The SISAL algorithm performs very well at extracting the endmembers for the i.i.d. dictionary $\boldsymbol{A}_1$, with N-FINDR occasionally grabbing the spotlight in figures 4.27 and 4.28. SISAL also performs very well for the $\boldsymbol{A}_2$ library.

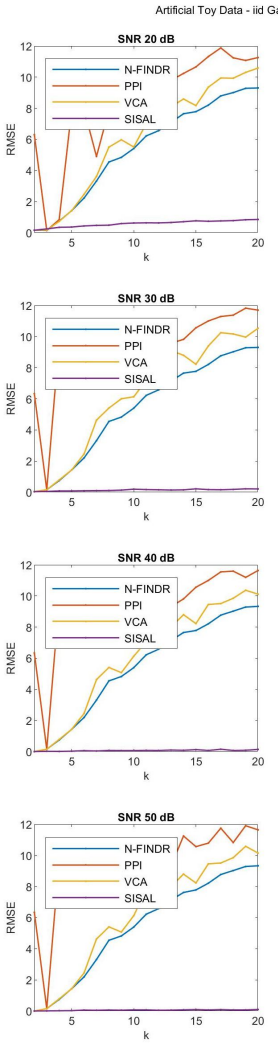


Fig. 4.27: EEAs performance for artificial toy data from $\boldsymbol{A}_1$.

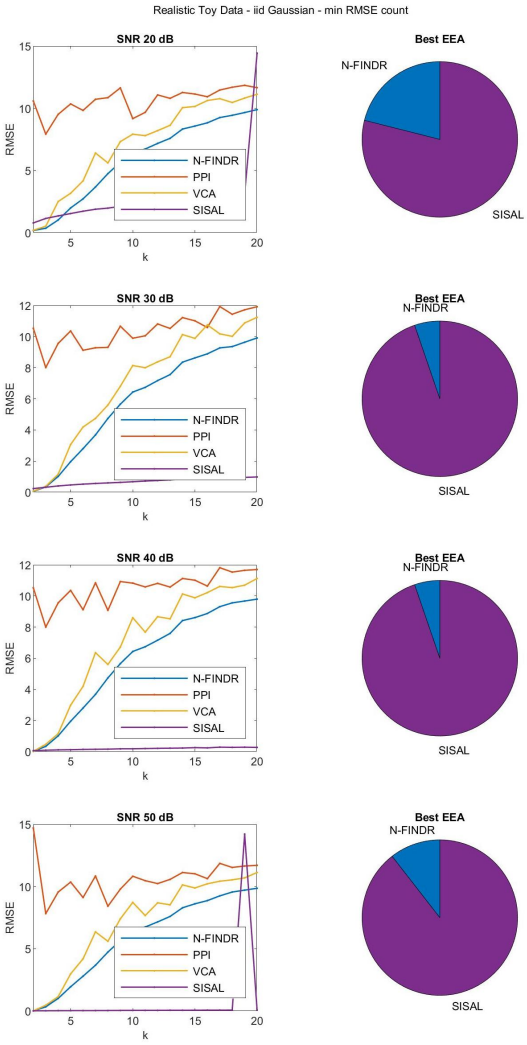
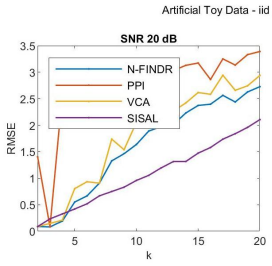


Fig. 4.28: EEAs performance for realistic toy data from $\boldsymbol{A}_1$.

It is interesting to note in figure 4.27 that the PPI did well just for $k = 3$, as the simulated data probably consisted of some pure pixels. N-FINDR and VCA also occasionally perform well.



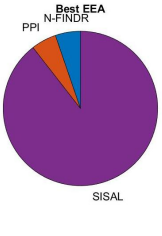
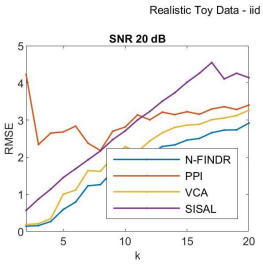


Fig. 4.29: EEAs performance for artificial toy data from $\boldsymbol{A}_2$.



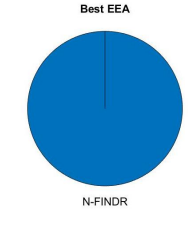
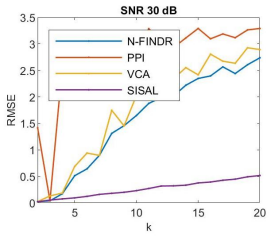
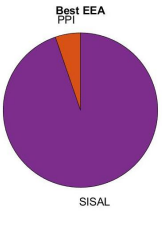
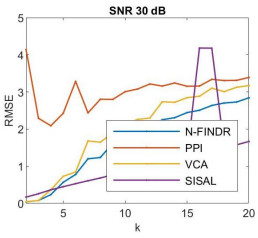
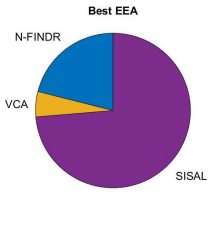


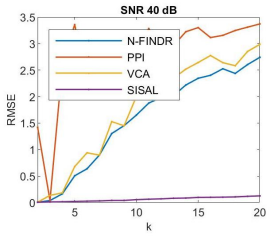
Fig. 4.30: EEAs performance for realistic toy data from $\boldsymbol{A}_2$.

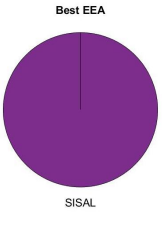


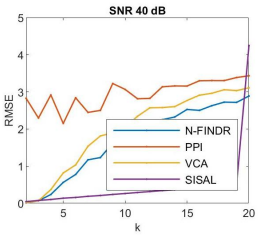


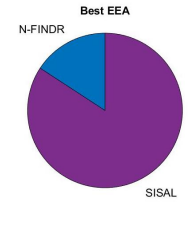


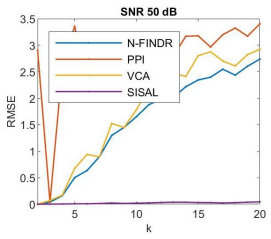


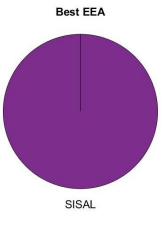


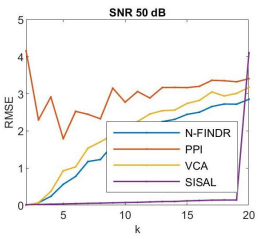


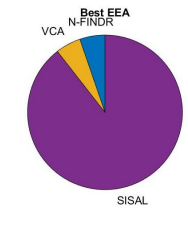












4.2.4 USGS vs USGS Pruned Libraries

There are no clear winners in figures 4.31 and 4.32 for the USGS 1995 spectral library. However the SISAL algorithm performs better when the data is less noisy even though it is supposed to work better with noisier data and outliers, while N-FINDR works better for noisy data. PPI does a little better for realistic toy data for larger k.



Fig. 4.31: EEAs performance for artificial toy data from $\boldsymbol{A}_3$.

Fig. 4.32: EEAs performance for realistic toy data from $\boldsymbol{A}_3$.

Similar performance for the pruned USGS library in figures 4.33 and 4.34, with SISAL extracting endmembers better for higher SNR, while N-FINDR performing better for noisier data. However, PPI features more since the library is pruned and thus signatures are less coherent.

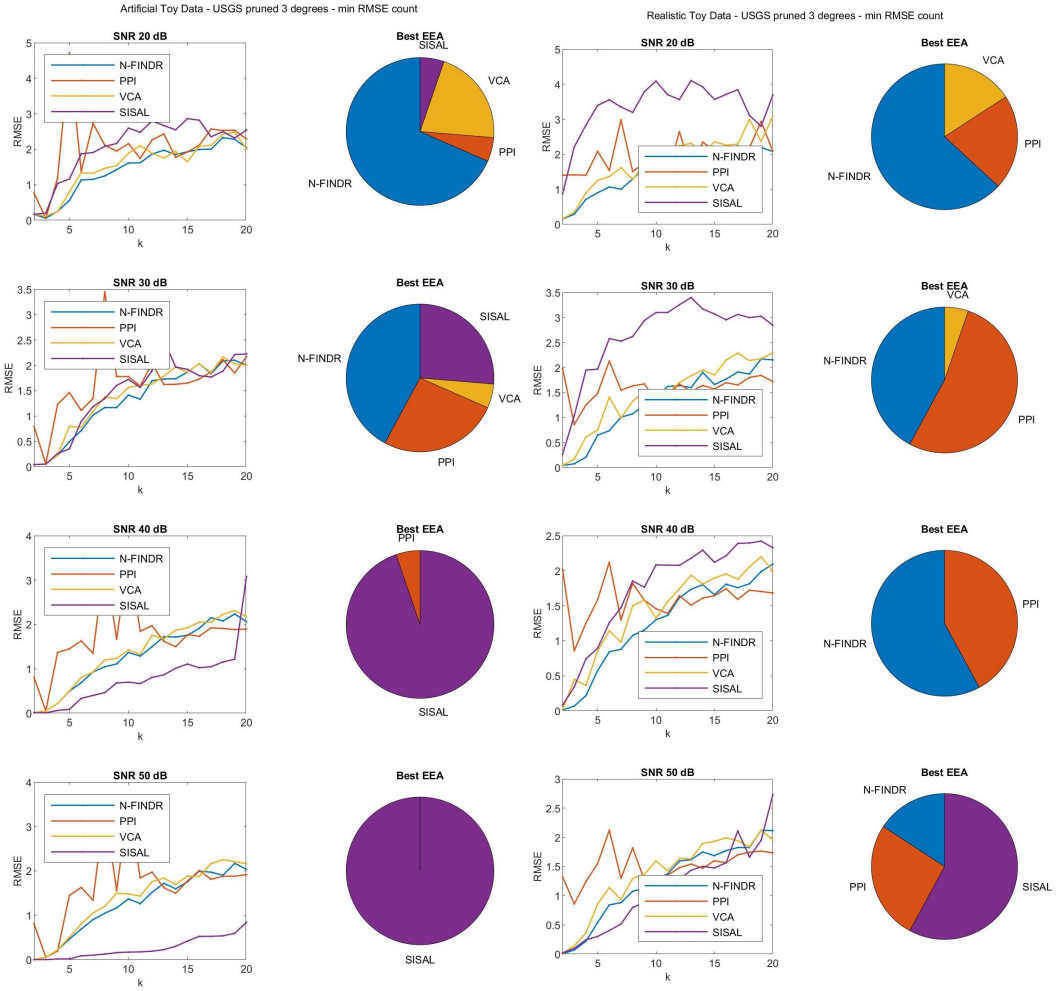


Fig. 4.33: EEAs performance for artificial toy data from $\boldsymbol{A}_4$.

Fig. 4.34: EEAs performance for realistic toy data from $\boldsymbol{A}_4$.

For the artificial images, the VCA performs consistently well on the Cuprite data, while N-FINDR works well on the Terrain data.

4.2.5 EEAs ComputationTimes

N-FINDR					**PPI**				
k	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	k	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$
2	0.0353	0.0326	0.0323	0.0372	2	0.0018	0.0014	0.0015	0.0037
3	0.0557	0.0400	0.0449	0.0419	3	0.0021	0.0019	0.0021	0.0022
4	0.0667	0.0499	0.0520	0.0490	4	0.0025	0.0022	0.0039	0.0023
5	0.0858	0.0614	0.0618	0.0650	5	0.0035	0.0026	0.0025	0.0028
6	0.0734	0.0751	0.0699	0.0695	6	0.0028	0.0029	0.0029	0.0029
7	0.0868	0.1072	0.0757	0.0804	7	0.0034	0.0044	0.0031	0.0032
...	...	...	...	...	...	...	...	...	...
14	0.2732	0.2849	0.1497	0.1966	14	0.0061	0.0053	0.0040	0.0047
15	0.3160	0.3054	0.1527	0.2394	15	0.0054	0.0058	0.0047	0.0050
16	0.3777	0.3964	0.1478	0.2695	16	0.0068	0.0060	0.0044	0.0050
17	0.8481	0.8316	0.1564	0.4296	17	0.0097	0.0073	0.0045	0.0053
18	0.9686	0.9885	0.1763	0.5415	18	0.0071	0.0069	0.0043	0.0066
19	1.1540	1.0803	0.1903	0.6384	19	0.0074	0.0073	0.0045	0.0058
20	1.2501	1.3052	0.2018	0.6350	20	0.0078	0.0100	0.0053	0.0063
VCA					**SISAL**				
k	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	k	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$
2	0.0054	0.0056	0.0026	0.0073	2	14.5598	31.2510	537.1011	46.3562
3	0.0090	0.0043	0.0057	0.0049	3	27.0457	30.4925	29.3117	9.5999
4	0.0053	0.0054	0.0073	0.0054	4	10.3440	15.5964	10.8696	6.5580
5	0.0123	0.0059	0.0055	0.0064	5	5.4268	8.1075	5.4936	7.4260
6	0.0064	0.0071	0.0064	0.0067	6	4.8158	5.7044	5.3834	6.3247
7	0.0080	0.0105	0.0080	0.0068	7	5.1120	5.4885	9.3972	7.6385
8	0.0084	0.0093	0.0072	0.0078	8	4.3847	5.6581	8.6878	6.2480
...	...	...	...	...	...	...	...	...	...
13	0.0121	0.0124	0.0092	0.0108	13	7.6636	4.8003	14.1535	10.6096
14	0.0124	0.0111	0.0093	0.0088	14	4.3703	5.1849	16.2018	9.0741
15	0.0125	0.0116	0.0100	0.0099	15	5.0320	4.9491	13.5610	11.6432
16	0.0139	0.0115	0.0103	0.0093	16	7.0616	9.9605	11.6879	11.4246
17	0.0179	0.0142	0.0104	0.0106	17	6.4267	7.9150	16.9613	10.9424
18	0.0147	0.0139	0.0083	0.0112	18	11.3808	7.6177	9.6906	11.7335
19	0.0149	0.0148	0.0093	0.0147	19	9.8100	7.8455	12.0488	13.3770
20	0.0147	0.0199	0.0096	0.0114	20	9.7743	8.7054	19.4589	8.4336

Tab. 4.2: Mean running times for extracting endmembers for artificial data

The trade-off lies clearly in the run-time of the EEAs. The following table 4.2 illustrates the time the EEAs would take to run on artificial toy data with $N = 10,000$ and for different sparsities k.

Table 4.2 shows that the SISAL computation times are very erratic, and it does take an awfully long time to fit a simplex around two endmembers, especially for the data simulated on the USGS library $\boldsymbol{A}_3$. The other geometrical extraction algorithms run well under 1 second, with VCA and PPI often running under 10 milliseconds, whose computational times do present an increasing trend with the increasing sparsity k, as expected.

4.3 Performance of Unmixing on A Priori Dictionaries v. Image-Extracted Dictionaries

Before discussing the performance discriminators for our experimental results on unmixing hyperspectral data based on standard dictionaries versus image-extracted dictionaries, first we present which algorithms we used and their notation going forward. Please note that these settings are the exact settings used by Bioucas-Dias in his MATLAB demonstrations.

- **BPDN** is the Basis Pursuit Denoising algorithm as presented in section 3.2.4 and algorithm 16 using Bioucas-Dias's `sunsal` MATLAB implementation with the parameter configuration: regularization parameter $\lambda = 0.002$, positivity constraint not enforced, and the sum-to-one constraint not enforced either.
- **CBPDN** is the Constrained BPDN algorithm as presented in in section 3.2.4 and algorithm 16 using Bioucas-Dias's `sunsal` MATLAB implementation with the parameter configuration: regularization parameter $\lambda = 0.002$, positivity constraint enforced, and only the sum-to-one constraint not enforced.
- **CLS** is the Constrained Least Squares algorithm as presented in section 3.2.4 and algorithm 15 using Bioucas-Dias's `sunsal` MATLAB implementation with the parameter configuration: no regularization, positivity constraint enforced, and only the sum-to-one constraint not enforced.
- **FCLS** is the Fully CLS algorithm as presented in section 3.2.4 and algorithm 15 using Bioucas-Dias's `sunsal` MATLAB implementation with the parameter configuration: no regularization parameter, and both the positivity constraint enforced, and the sum-to-one constraint enforced.

- **OMP** is the classic greedy Orthogonal Matching Pursuit algorithm as presented in section 3.2.3 and algorithm 11 using Becker's `omp` MATLAB implementation with configuration paramenter for estimated sparsity set to $\hat{k}$.
- **CoSaMP** is the Compressive Sampling Matching Pursuit algorithm as presented in section 3.2.3 and algorithm 12 using Becker's `omp` MATLAB implementation with configuration parameter for estimated sparsity set to $\hat{k}$.

Let us remind ourselves of the research question that we are trying to answer in this section. Do spectral dictionaries available *a priori* perform better than image-extracted dictionaries in reconstructing the image? What is the optimal sparsity of signals which are recoverable using hyperspectral libraries versus image-extracted dictionaries? Thus we need a measure that indicates the success of reconstruction.

Instead of the classic RMSE measure, Iordache [9] uses the following performance discriminators: *signal-to-reconstruction error* (SRE) and *probability of success* p_s. The SRE is defined as follows in (4.4):

$$SRE = E[\|\boldsymbol{x}\|_2^2]/E[\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2] \tag{4.4}$$

where the expected values in the nominator and denominator are the sample averages. This ratio gives us more information in terms of the power of the signal $\boldsymbol{x}$ in relation to the power of the error, where as the classic RMSE measure only considers the error alone, and not the energy of the signal. The measure given in decibels is $SRE(dB) = 10\log_{10}(SRE)$. Although this relative measure is much more useful than the RMSE, as it gives us a sense of the relation between power of the signal and of the error, we cannot say with certainty what a good SRE measure is. That is why Iordache uses the probability of success given by equation (4.5):

$$p_s = P\left(\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2/\|\boldsymbol{x}\|_2^2 \leq 1/\tau\right) \tag{4.5}$$

which is an estimate of the probability whether the relative error power is smaller than a threshold value $1/\tau$. This can be approximated as $p_s = P\left(1/SRE \leq 1/\tau\right)$, i.e. making sure that the relative error power $1/SRE$ is smaller than the threshold. This gives us a better sense if the reconstruction is good enough. Iordache sets the threshold $\tau = 3.16$, i.e. $10\log_{10}(3.16) = 5\,dB$ which he considers a more stable estimate rather than setting $\tau = 10$ which would give a probability of success of $p_s = 1$ if the relative error of the fractional abundances was less than $1/10$. Thus reconstruction is considered successful for $1/SRE \leq 1/3.16 \approx 0.32$, or alternatively for the signal-to-reconstruction error ratio $SRE \geq 3.16$. For a successful recovery of

the original hyperspectral signal, we seek $SRE(dB) \geq 5\,dB$.

We analyzed the unmixing algorithms mentioned earlier on the various dictionaries for $SNR = 30\,dB$ as that is the common signal-to-noise ratio for testing in published papers [9]. The probability of success for the i.i.d. dictionaries $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ is consistently $p_s = 1$. We are more concerned with the USGS library, as the random i.i.d. libraries are more useful for proof of concept. For our specific MATLAB configurations for $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$, unmixing using the `sunsal` MATLAB implementation, the FCLS algorithm performed the best, with CLS and CBPDN sharing second place.

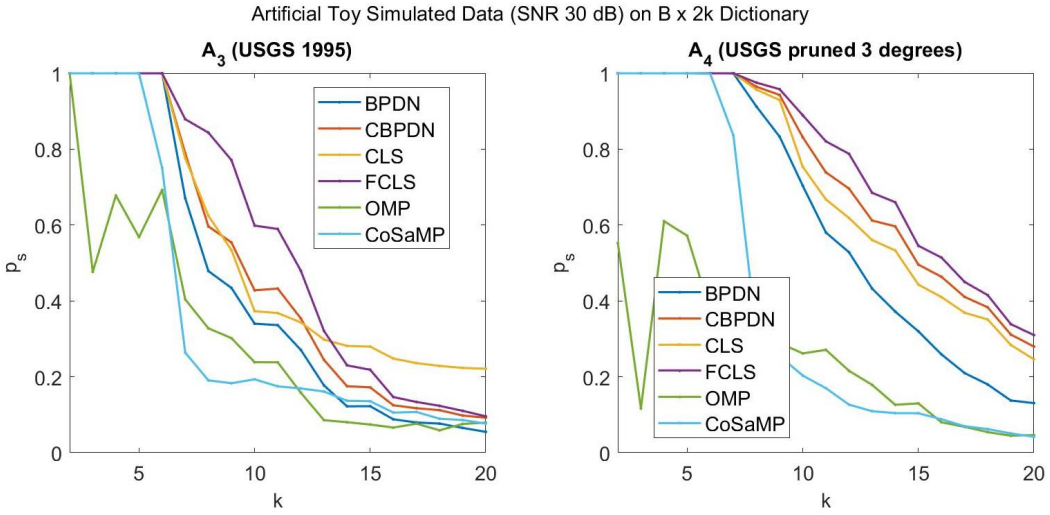


Fig. 4.35: p_s results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on given dictionaries $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$.

The key here is that enforcing the constraints results in better performance. In our experiments, it looks like $k \leq 6$ for $\boldsymbol{A}_3$ and $k \leq 7$ for $\boldsymbol{A}_4$ gives us $p_s = 1$ for most sparse regression algorithms with our specific experiment configurations, see figure 4.35. The worst performing are the greedy OMP and CoSaMP algorithms.

We can see in figure 4.36 that the FCLS algorithm has $SRE(dB) > 5\,dB$ for $k \leq 12$ using dictionary $\boldsymbol{A}_3$, while $k \leq 16$ yields in good performance for $\boldsymbol{A}_4$. However the larger k the more endmembers the algorithms have to unmix, and thus yield poorer recovery of the original signal, in our case.

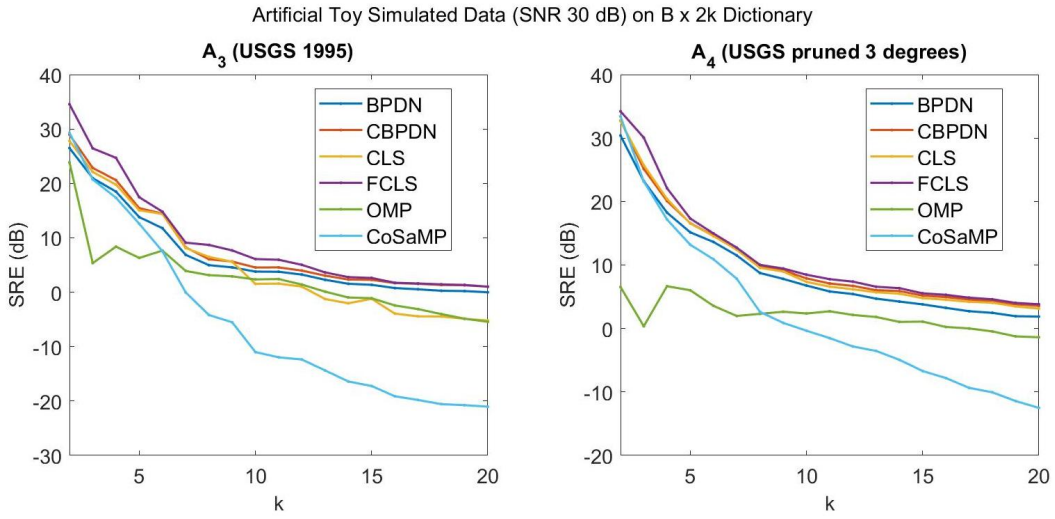


Fig. 4.36: SRE (dB) results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on given dictionaries $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$.

The key points that we can take from this are that SRE gets worse with increasing the number of endmembers present in a scene, and that enforcing the non-negativity and sum-to-one constraints yield best reconstruction results. Also as expected, the SRE gets better with improving SNR as seen in figure 4.37, except for the OMP.

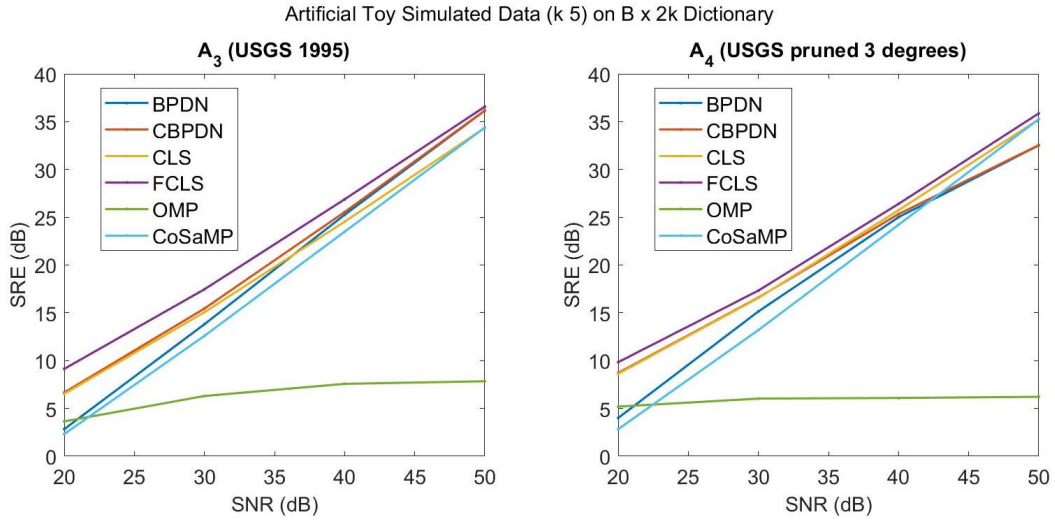


Fig. 4.37: SRE (dB) versus SNR (dB) comparison for various unmixing algorithms on $\boldsymbol{A}_3$ and $\boldsymbol{A}_4$.

So far, these algorithms behaved in a similar fashion to Iordache's results [9]. Next we take a look at how the unmixing algorithms worked for unmixing the artificial toy data using image-extracted material matrices. The way we chose the image-extracted dictionaries was to choose the best EEA algorithm with the minimum RMSE from the previous section 4.2 for the various k and SNR configurations, and thus select those best extracted endmembers. We denote these image-extracted dictionaries by $\boldsymbol{M}$ to differentiate from the available libraries a priori denote with $\boldsymbol{A}$. We do not consider the underperforming OMP and CoSaMP algorithms for the remainder of this paper.

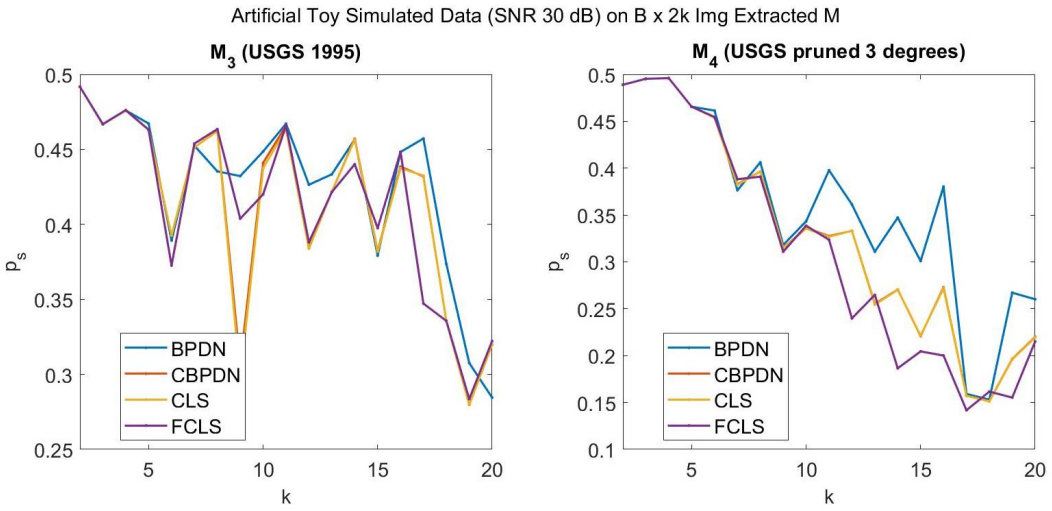


Fig. 4.38: p_s results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on image-extracted dictionaries $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$.

Unmixing based on image-extracted dictionaries yielded poorer reconstruction with less than 50% probability of success for all k's with our configurations, see figure 4.38. There is no optimal number of image-extracted endmembers that will yield $SNR(dB) > 5\,dB$ unfortunately as seen in figure 4.39.

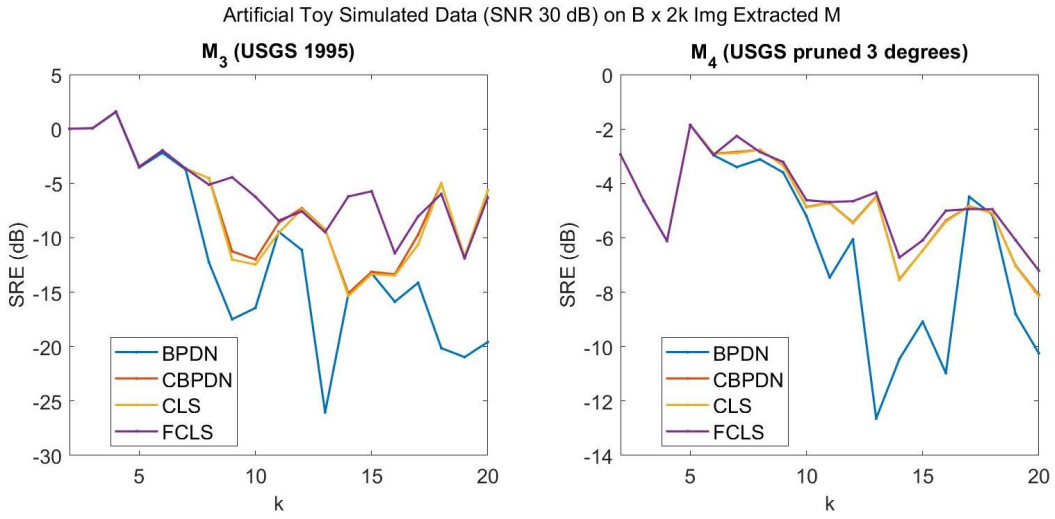


Fig. 4.39: SRE (dB) results obtained on the unmixing artificial toy data for $SNR = 30\,dB$ using various unmixing algorithms on image-extracted mixing matrices $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$.

From the probability of success calculations, we observe that unmixing hyperspectral data using dictionaries available a priori performed better than unmixing on the image-extracted mixing matrices.

`sunsal` finds the optimal subset of fractional abundances faster on image-extracted dictionaries, since there are less endmembers to choose from, as in table 4.3. This small time advantage is compromised by the time taken to extract the endmembers using an EEA.

BPDN					**CBPDN**				
k	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	k	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$
2	0.0801	0.0707	0.1789	0.0896	2	0.1251	0.1152	0.1973	0.2610
3	0.1353	0.1486	0.1190	0.1314	3	0.2111	0.2582	0.1779	0.1985
4	0.1476	0.1722	0.1193	0.1280	4	0.2599	0.2808	0.1667	0.2058
5	0.1725	0.2114	0.1705	0.1422	5	0.3741	0.3336	0.2026	0.2314
6	0.1893	0.2729	0.1704	0.2036	6	0.3809	0.4017	0.2353	0.2964
7	0.1994	0.1970	0.1873	0.1617	7	0.4639	0.3891	0.3047	0.3044
8	0.2686	0.2216	0.2527	0.1740	8	0.6098	0.4637	0.4029	0.3478
...	...	...	...	...	...	...	...	...	...
13	0.4730	0.3539	0.3110	0.3226	13	0.9972	0.7093	0.7661	0.5344
14	0.5086	0.3836	0.3752	0.3398	14	1.1536	0.8027	0.8804	0.5364
15	0.6370	0.4028	0.4376	0.3688	15	1.1638	0.8543	0.7120	0.5708
16	0.6301	0.4628	0.4562	0.3800	16	1.4033	0.8907	0.8236	0.6536
17	0.6710	0.4954	0.4617	0.4162	17	1.5191	0.9686	0.7629	0.6590
18	0.6969	0.4904	0.5023	0.4038	18	1.5848	0.9755	1.0141	0.7070
19	0.7919	0.5344	0.4734	0.4274	19	1.7532	1.0238	1.1174	0.7530
20	0.8805	0.5891	0.5727	0.4522	20	1.8577	1.1523	1.1404	0.7826
BPDN					**CBPDN**				
k	$\boldsymbol{M}_1$	$\boldsymbol{M}_2$	$\boldsymbol{M}_3$	$\boldsymbol{M}_4$	k	$\boldsymbol{M}_1$	$\boldsymbol{M}_2$	$\boldsymbol{M}_3$	$\boldsymbol{M}_4$
2	0.0507	0.0444	0.1410	0.0539	2	0.0517	0.0442	0.1339	0.0536
3	0.0609	0.0596	0.0611	0.0591	3	0.0817	0.0897	0.0874	0.1011
4	0.0875	0.0708	0.0705	0.0725	4	0.0807	0.0934	0.1168	0.0977
5	0.0795	0.0780	0.0777	0.0944	5	0.0928	0.0841	0.0987	0.1112
6	0.0901	0.0900	0.0821	0.0882	6	0.0881	0.0931	0.1456	0.1844
7	0.0973	0.1038	0.1235	0.0941	7	0.1004	0.0997	0.1616	0.2025
8	0.1346	0.1417	0.1293	0.1056	8	0.1681	0.1239	0.1407	0.2153
...	...	...	...	...	...	...	...	...	...
13	0.2424	0.1725	0.1488	0.1610	13	0.2423	0.2061	0.2694	0.2362
14	0.2087	0.1464	0.1566	0.1022	14	0.2146	0.1805	0.3280	0.2301
15	0.2349	0.1398	0.1392	0.1233	15	0.2467	0.1929	0.4107	0.3062
16	0.2627	0.1660	0.1345	0.1079	16	0.2660	0.1934	0.3954	0.2218
17	0.2781	0.1669	0.1812	0.1086	17	0.2789	0.2165	0.3581	0.2539
18	0.2999	0.1947	0.1078	0.1095	18	0.3291	0.2285	0.2550	0.2393
19	0.3396	0.2038	0.0965	0.1126	19	0.3329	0.2465	0.2680	0.2467
20	0.3808	0.2093	0.0917	0.1299	20	0.3657	0.2526	0.2321	0.4797

Tab. 4.3: Mean running times unmixing on available *a priori* v. image-extracted dictionaries

4.4 Performance of Reconstructing Various Data

In this section we present visually how well sparse unmixing can reconstruct a hyperspecrtal image. We will focus mainly on the spatially correlated artificial Cuprite and Terrain images, as these make a lot more sense to visualise rather than randomly generated images that are not spatially correlated. A reminder that the artificial Cuprite and Terrain images were generated from the actual images described in section 4.1, but first using the N-FINDR EEA to extract k endmembers and FCLS to work out the fractional abundance maps, then synthesised back together with only k endmembers.

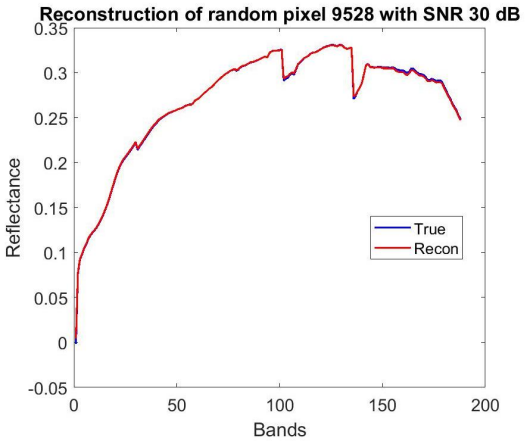


Fig. 4.40: Random Cuprite pixel 9528 reconstructed spectrum

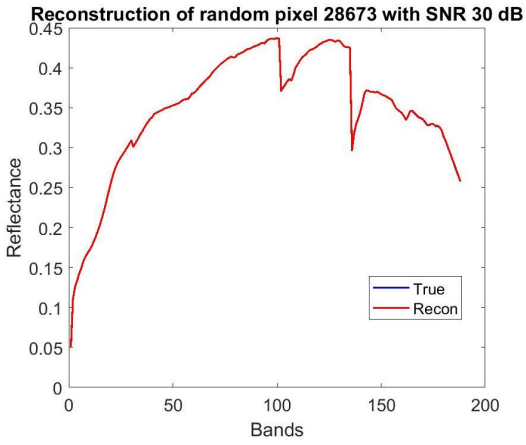


Fig. 4.41: Random Cuprite pixel 28673 reconstructed spectrum

Figures 4.40 and 4.41 show the reconstructed spectrum versus the true spectrum for two randomly selected pixels out of the artificial Cuprite image with $k = 10$. For this particular scenario we obtained $RMSE = 19.1$, $SRE = 17.2\,dB$ and $p_s = 1$. Visually we can see how close the reconstructed spectrums look to the original with the probability of success of 1.

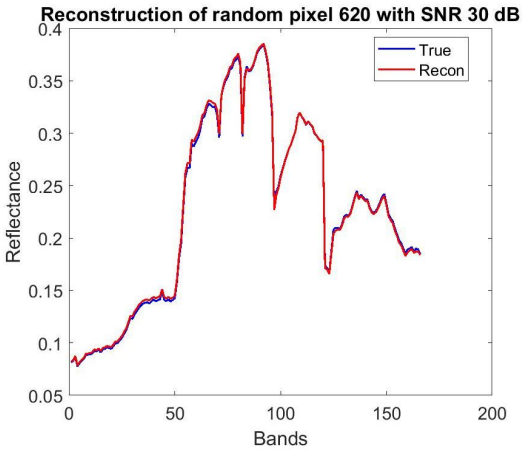


Fig. 4.42: Random Terrain pixel 620 reconstructed spectrum

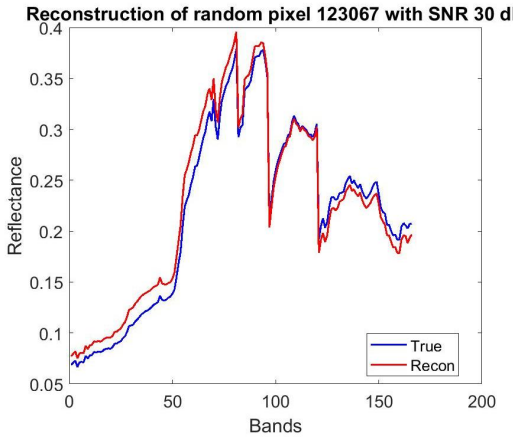


Fig. 4.43: Random Terrain pixel 123067 reconstructed spectrum

Figures 4.42 and 4.43 show the reconstructed spectrum versus the true spectrum for two randomly selected pixels out of the artificial Terrain image with $k = 10$. For this particular scenario we obtained $RMSE = 58.4$, $SRE = 10.7\,dB$ and $p_s = 0.86$. The probability of success being slightly less than 1 tells us that the reconstruction will be close but some room for error, which is illustrated in these two figures, especially for pixel 123067.

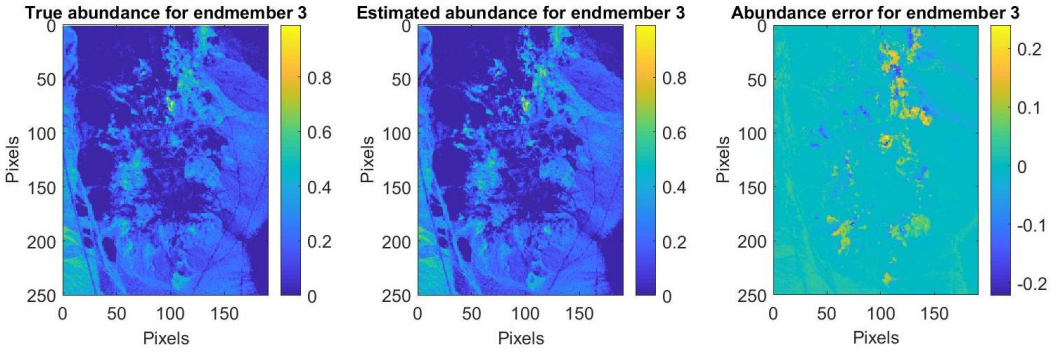


Fig. 4.44: Comparison of the third endmember's fractional abundance map of the Cuprite image.

Comparing the true abundance map against the estimated abundance map for randomly selected endmember 3 in figure 4.44, we see the estimation error varies between -0.2 and 0.2 although occurring mainly between 0 and 0.1. We cannot infer much from the abundance error as it is relative to the endmember it represents.

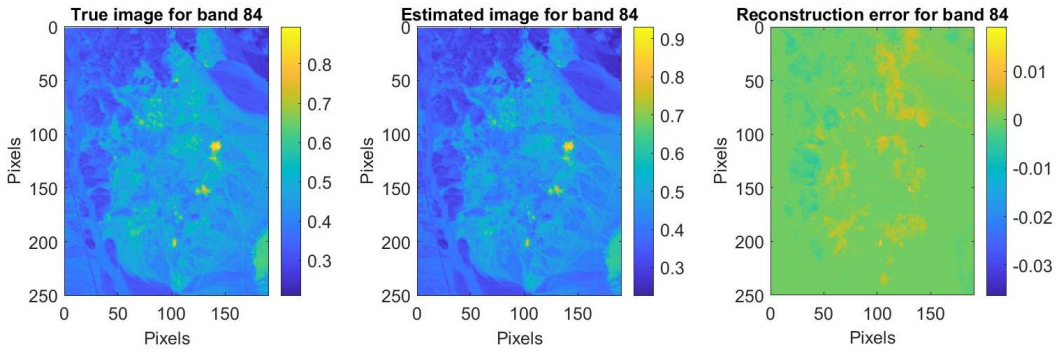


Fig. 4.45: Comparison of the reconstructed spectral band 84 of the Cuprite image.

More importantly, the comparison of the randomly selected band 84 in figure 4.45 shows the reconstruction error varying very closely to 0. This is to be expected for a probability of success equal to 1.

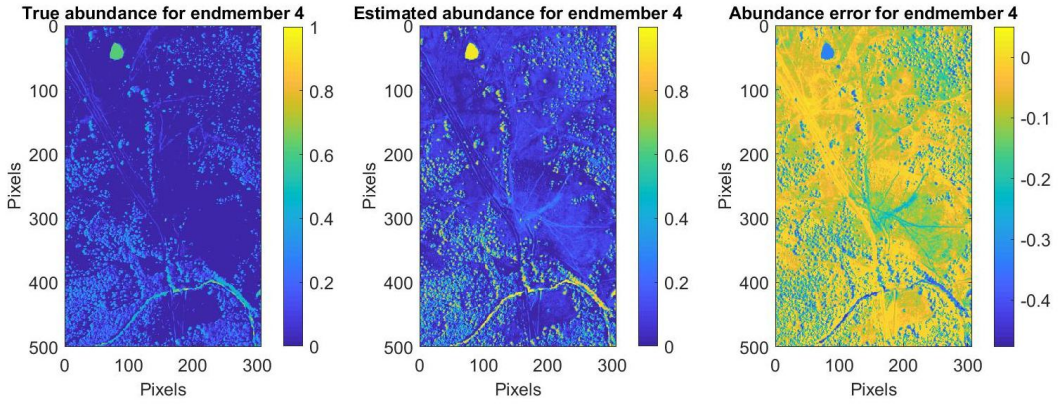


Fig. 4.46: Comparison of the fourth endmember's fractional abundance map of the Terrain image.

Figure 4.46 shows the comparison of the fractional abundance for randomly selected endmember 4 of the Terrain image. The estimation error does vary a lot more between -0.4 and 0.

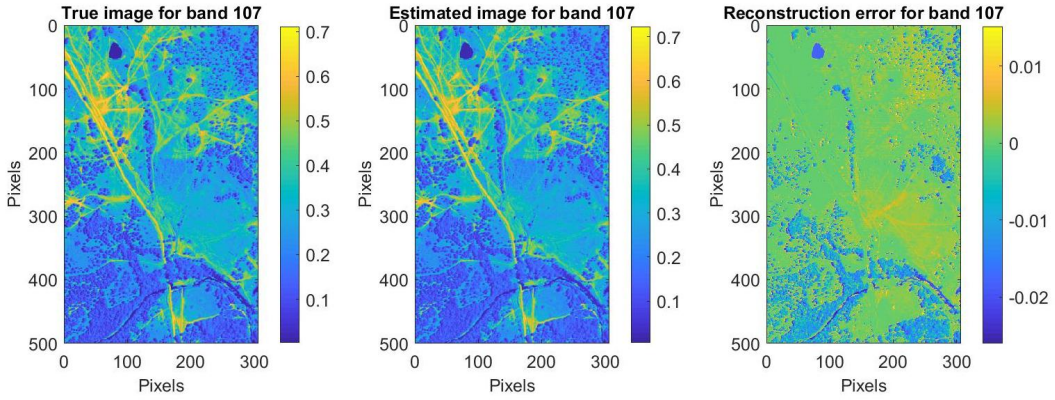


Fig. 4.47: Comparison of the reconstructed spectral band 107 of the Terrain image.

Figure 4.47 for Terrain's spectral band 107 shows reconstruction error varying mainly between -0.02 and 0.01 which is not bad considering the probability of success $p_s = 0.86$. It is very difficult to see the reconstruction differences with the naked eye, as the estimated image looks so remarkably close to the original.

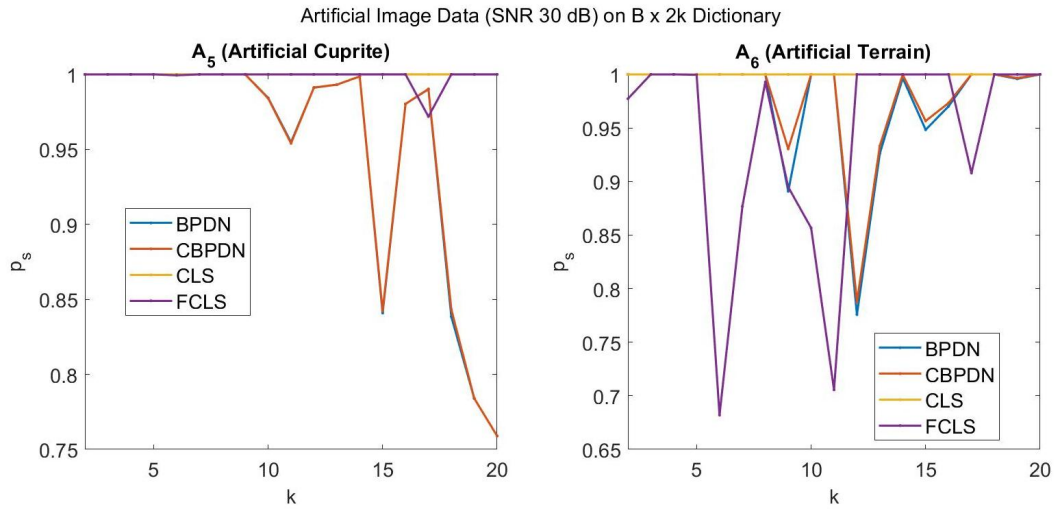


Fig. 4.48: p_s plot for the artificial Cuprite and Terrain images at $SNR = 30\,dB$.

The probability of success might look chaotic in figure 4.48 for $k > 9$ on the left, and $k > 5$ on the right, but note that $p_s > 0.75$ for the Cuprite image and $p_s > 0.65$ for Terrain image. Thus reconstruction is quite successful for these images, and most interesting the CLS algorithm performs superbly for both of these images with

consistent $p_s = 1$.

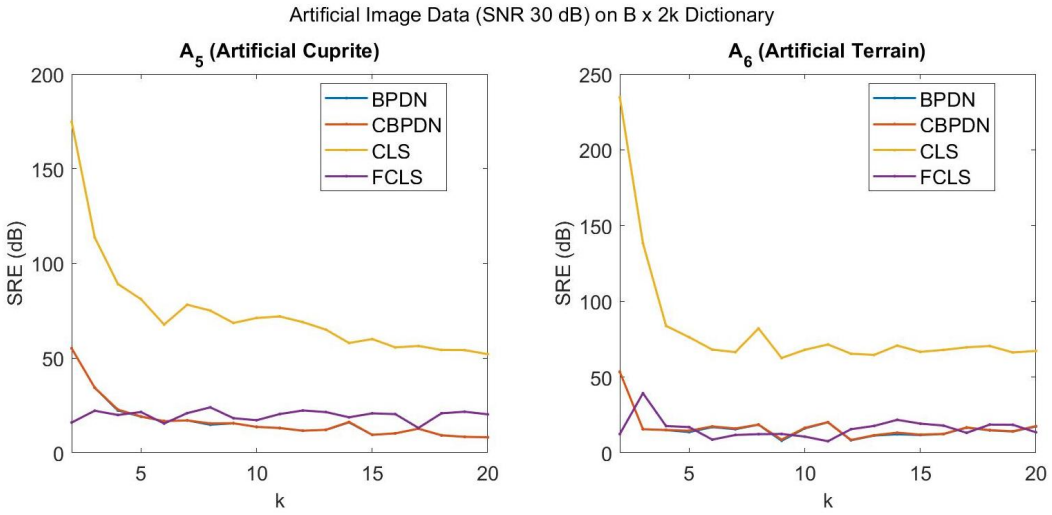


Fig. 4.49: SRE plot for the artificial Cuprite and Terrain images at $SNR = 30\,dB$.

Figure 4.49 clearly shows that the CLS algorithm outperforms the other algorithms by far with quite SRE values. Bioucas-Dias's MATLAB implementation of `sunsal` performs very well reconstructing hyperspectral images in any of the four configurations it was designed for.

4.5 Summary of Results

The key points to take from these experiments are:

- For artificial and realistic toy data, SISAL seems to extract the endmembers the best for most libraries, with N-FINDR as second best, for the high SNR setting. VCA and N-FINDR worked well for the artificial Cuprite and Terrain images. Although we cannot generalise which EEA works best in all situations, the RMSE measure helped us decide on the appropriate EEA.

- Unmixing on the given dictionaries a priori outperformed unmixing on the image-extracted dictionaries for our configurations. However, there is a compromise in terms of running time. Unmixing using image-extracted dictionaries does run faster. We do need to take in account the time endmember extraction time though.

- In terms of reconstructing hyperspectral images, the probability of success is an appropriate measure. In general, the higher the number of endmembers to unmix, the less accurate the reconstruction. Enforcing the non-negativity constraint definitely improved unmixing accuracy. Sum-to-one did not always help the situation. The greedy algorithms were poor performers.

Chapter 5

Conclusion and Further Research

Our main goal was to provide an extensive literature review on the existing geometrical methods for extracting endmembers from hyperspectral images, and on the sparse regression methodologies for solving the unmixing problem of the fractional abundances. We believe this goal was achieved in chapter 2 in detail. As Bioucas-Dias presents his findings in [30], the number of publications on hyperspectral data has increased yearly comparable to publications on radar in the remote sensing field. Thus we had abundant literature to draw from, even though very few publications present the whole unmixing process in such detail as this report.

In chapter 3 we set out to present the pseudo-code and additional mathematical detail for all the algorithms used for our computer simulation experiments in chapter 4 which aimed at answering our research questions. We had to be selective with the results presented in chapter 4 as we ran 6232 scenarios in MATLAB. Due to the time frame of this research report, we could only allow ourselves to work on simulated data as per Bioucas-Dias's previous experiments, and we had to be innovative about how we simulate artificial data in new ways. We have achieved that by randomly mixing the endmembers per pixel, not just the scene, as in reality we don't find the exact same 4 or 5 materials in every single pixel. The scene might be limited to 20 to 25 materials. Each pixel will have 4 or 5 very different materials out of those 20 to 25 endmembers in the scene. The only problem with these "realistic" toy examples is that the pixels are not spatially correlated. This was the reason for employing the frequently used images of the Cuprite and Terrain scenes, and extracting a certain number of endmembers which were sythesized to form new artificial images.

5.1 Conclusion

In conclusion, our sincere hope is that this comparative analysis achieves the main goal of providing the full picture of hyperspectral linear unmixing for any academic and professional interested in this topic. Right from the outset, the aim for this paper has been to bridge the gap in the literature by providing a rich resource of the end-to-end process of unmixing hyperspectral images. It is such an important and intricate process that enables material classification, resolution enhancement, and hyperspectral data compression. For more specialised literature see the Bibilography following after this chapter.

In attempting to answer our research questions, we found that it is efficient to identify the lower-dimensional subspace in which the data actually lives, as Bioucas-Dias suggested. By reducing the high dimensionality of the hyperspectral data, we estimate the intrinsic dimension of the data. This sparsity estimate is mandatory to running the endmember extraction algorithms (EEAs), if no spectral library is available *a priori* and we need to extract these spectral signatures from the image. We only chose 4 endmember extraction algorithms out of the multitude of algorithms available, that we thought would best represent the different geometrical approaches in the literature. Even though the PPI and N-FINDR algorithms are still the most commonly used EEAs, there are better performing algorithms that take care of outliers and noise. The EEAs are still a matter of preference with many professionals.

We focused on the linear mixing model throughout this paper. The greatest benefit of using sparse unmixing is that we can take advantage of the increasingly available spectral libraries of materials measured on the ground, thus bypassing endmember extraction algorithms. Linearly unmixing on available libraries with spectral signatures known in advance performed better than on image-extracted dictionaries with our specific configurations as shown in chapter 4 given the compromises in running time. These experiments do not provide us with the authority to deduce which method works best in practice. Professionals in the remote sensing industry still use image-extracted dictionaries for unmixing. The problem with unmixing on available spectral libraries is that laboratory measured spectral signatures are not exactly the same as remote sensed data affected by atmospheric interferences. On the other hand, unmixing on image-extracted dictionaries runs the risk of missing out some spectral signatures due to noisy data, and the extracted endmembers might not have physical relevance. In conclusion, estimating fractional abundance maps for linearly combined endmembers no longer depends on the EEAs performance, and can be bypassed if the relevant spectral libraries are available.

5.2 Future Research

Our experiments only explored simulated toy data and artificial Cuprite and Terrain data due to the restrictive nature of this report. Experiments on real data other than Cuprite would be of great interest, however prior knowledge of the ground truth would remain a challange. Identifying the intrinsic dimensionality of the lower dimensional subspace in which the high dimensional data actually lives is of great interest as well. Alternatives to the HySime algorithm have been published, which are worthy of further investigation, and likewise, improvements on "realistic" hyperspectral data simulations.

Faster and more accurate unmixing algorithms have been developed all mainly spawning from the augmented Lagrangian method of multipliers. This is due to the high correlation between spectral signatures in a given dictionary, which is usually balanced by the very sparse nature of the fractional abundances. The current trend is to use machine learning for learning dictionaries from images, alternative unmixing methods, data compression, and target identification, which are all worthy of further research.

Spectral unmixing depends on the intrinsic dimension, i.e. the sparsity of the data, and not the number of pixels in the hyperspectral image. Since pixels are basically analysed individually, not the whole scene at once, this implies parallel nature for the processing algorithms. Thus hyperspectral images analysis would benefit greatly from dividing an image into sub-images, i.e. subsets of pixels, and by processing these sub-images in parallel. This would accelerate processing and provide high scalability.

Engineers have had to take their own initiative to implement hyperspectral unmixing algorithms, as there is not a hyperspectral processing toolbox that caters for this purpose in MATLAB. With the advent of the data science research field, open-source languages such as Python and R are establishing themselves as the data scientist's favoured working tools. Even though remote sensing falls under electrical engineering, geosciences and applied mathematics, it is undeniably a data-driven analysis field. Thus Python projects such as SPy 2016, SPy 2017 and HypPy, and R projects HyperSpec and hsdar have been developed in these data science tools. Our hope is that new advanced hyperspectral unmixing algorithms will be included in future releases.

Bibliography

[1] Mark R Elowitz. What is imaging spectroscopy (hyperspectral imaging)? *Retrieved November*, 27:2013, 2013.

[2] Gary A Shaw and Hsiaohua K Burke. Spectral imaging for remote sensing. *Lincoln laboratory journal*, 14(1):3–28, 2003.

[3] Jakub Bieniarz. *Sparse methods for hyperspectral unmixing and image fusion.* Doctorate diss., Universität Osnabrück, 2015.

[4] Naoto Yokoya, Claas Grohnfeldt, and Jocelyn Chanussot. Hyperspectral and multispectral data fusion: A comparative review of the recent literature. *IEEE Geoscience and Remote Sensing Magazine*, 5(2):29–56, 2017.

[5] José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, 2010.

[6] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[7] Mauro Dalla Mura. An overview on hyperspectral unmixing. `http://www-ljk.imag.fr/membres/Faouzi.Triki/projetPbsInverses/Pre/DallaMura_unmixing.pdf`, 2014. Presentation, GIPSA-lab, Grenoble-INP.

[8] Richard G Baraniuk. Compressive sensing [lecture notes]. *IEEE signal processing magazine*, 24(4):118–121, 2007.

[9] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):2014–2039, 2011.

[10] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[11] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

[12] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[13] Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, 21(1):130–144, 2012.

[14] Xiaoxia Sun, Nasser M Nasrabadi, and Trac D Tran. Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4457–4471, 2015.

[15] Soheil Bahrampour, Nasser M Nasrabadi, Asok Ray, and William Kenneth Jenkins. Multimodal task-driven dictionary learning for image classification. *IEEE transactions on Image Processing*, 25(1):24–38, 2016.

[16] Marian-Daniel Iordache. *A sparse regression approach to hyperspectral unmixing*. Doctorate diss., Instituto Superior Técnico, Department of Electrical and Computer Engineering, 2011.

[17] Miguel A Veganzones, Guillaume Tochon, Mauro Dalla-Mura, Antonio J Plaza, and Jocelyn Chanussot. Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation. *IEEE Transactions on Image Processing*, 23(8):3574–3589, 2014.

[18] Lucas Drumetz. *Endmember variability in hyperspectral image unmixing*. Doctorate diss., Université de Grenoble Alpes, GIPSA-lab, 2016.

[19] Maofeng Tang, Lianru Gao, Andrea Marinoni, Paolo Gamba, and Bing Zhang. Integrating spatial information in the normalized p-linear algorithm for nonlinear hyperspectral unmixing. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(4):1179–1190, 2018.

[20] Shutao Li, Renwei Dian, Leyuan Fang, and José M Bioucas-Dias. Fusing hyperspectral and multispectral images via coupled sparse tensor factorization. *IEEE Transactions on Image Processing*, 27(8):4118–4130, 2018.

[21] Xiang Xu, Jun Li, Changshan Wu, and Antonio Plaza. Regional clustering-based spatial preprocessing for hyperspectral unmixing. *Remote Sensing of Environment*, 204:333–346, 2018.

[22] Chia-Hsiang Lin and José M Bioucas Dias. New theory for unmixing ill-conditioned hyperspectral mixtures. In *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 430–434. IEEE, 2018.

[23] Jihao Yin, Chenyu Huang, Xiaoyan Luo, and Qian Du. Automatic endmember bundle unmixing methodology for lunar regional area mineral mapping. *Icarus*, 319:349–362, 2019.

[24] Yan Li, Shaoquan Zhang, Chengzhi Deng, and Shengqian Wang. Reweighted local collaborative sparse regression for hyperspectral unmixing. *Infrared Physics & Technology*, 2019.

[25] Dongqing Li, Xuesong Wang, and Yuhu Cheng. Spatial-spectral neighbour graph for dimensionality reduction of hyperspectral image classification. *International Journal of Remote Sensing*, pages 1–23, 2019.

[26] Antonio Plaza, Jon Atli Benediktsson, Joseph W Boardman, Jason Brazile, Lorenzo Bruzzone, Gustavo Camps-Valls, Jocelyn Chanussot, Mathieu Fauvel, Paolo Gamba, Anthony Gualtieri, et al. Recent advances in techniques for hyperspectral image processing. *Remote sensing of environment*, 113:S110–S122, 2009.

[27] Alexander FH Goetz, Gregg Vane, Jerry E Solomon, and Barrett N Rock. Imaging spectrometry for earth remote sensing. *Science*, 228(4704):1147–1153, 1985.

[28] Hairong Wang and Turgay Celik. Sparse representation-based hyperspectral image compression and classification. *Signal, Image and Video Processing*, 12(5):1009–1017, 2018.

[29] Nasser M Nasrabadi. Hyperspectral target detection: An overview of current and future challenges. *IEEE Signal Processing Magazine*, 31(1):34–44, 2014.

[30] José M Bioucas-Dias, Antonio Plaza, Gustavo Camps-Valls, Paul Scheunders, Nasser Nasrabadi, and Jocelyn Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and remote sensing magazine*, 1(2):6–36, 2013.

[31] Gustavo Camps-Valls, Devis Tuia, Lorenzo Bruzzone, and Jón Atli Benediktsson. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *IEEE Signal Processing Magazine*, 31(1):45–54, 2014.

[32] Robert O Green, Michael L Eastwood, Charles M Sarture, Thomas G Chrien, Mikael Aronsson, Bruce J Chippendale, Jessica A Faust, Betina E Pavri, Christopher J Chovit, Manuel Solis, et al. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (aviris). *Remote sensing of environment*, 65(3):227–248, 1998.

[33] Steven M Bergma. *The utility of hyperspectral data to detect and discriminate actual and decoy target vehicles.* Doctorate diss., Naval Postgraduate School, Monterey, California, 1996.

[34] Md Aktaruzzaman. Simulation and correction of spectral smile effect and its influence on hyperspectral mapping. ITC, 2008.

[35] Naoto Yokoya, Norihide Miyamura, and Akira Iwasaki. Preprocessing of hyperspectral imagery with consideration of smile and keystone properties. In

Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques, and Applications III, volume 7857, page 78570B. International Society for Optics and Photonics, 2010.

[36] Xavier Ceamanos and Sylvain Douté. Spectral smile correction of crism/mro hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):3951–3959, 2010.

[37] Harris Geospatial Solutions Inc. Preprocessing aviris data tutorial. `http://www.harrisgeospatial.com/docs/PreprocessAVIRIS.html`, 2018.

[38] Harris Geospatial Solutions Inc. Radiometric caliberation. `http://www.harrisgeospatial.com/docs/RadiometricCalibration.html`, 2018.

[39] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing, 3rd ed.* Pearson, 2008.

[40] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013.

[41] Emmanuel J Candès and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[42] Emmanuel Candès, Mark Rudelson, Terence Tao, and Roman Vershynin. Error correction via linear programming. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 668–681. IEEE, 2005.

[43] David L Donoho and Jared Tanner. Sparse nonnegative solution of underdetermined linear equations by linear programming. *Proceedings of the National Academy of Sciences*, 102(27):9446–9451, 2005.

[44] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[45] Emmanuel J Candès, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[46] Emmanuel J Candès et al. Compressive sampling. In *Proceedings of the international congress of mathematicians*, volume 3, pages 1433–1452. Madrid, Spain, 2006.

[47] Nirmal Keshava and John F Mustard. Spectral unmixing. *IEEE signal processing magazine*, 19(1):44–57, 2002.

[48] Grégoire Mercier and Marc Lennon. Support vector machines for hyperspectral image classification with spectral-based kernels. In *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, volume 1, pages 288–290. IEEE, 2003.

[49] Javier Plaza, Pablo Martínez, Rosa Pérez, and Antonio Plaza. Nonlinear neural network mixture models for fractional abundance estimation in aviris hyperspectral images. In *Proc. XIII JPL Airborne Earth Science Workshop*, 2004.

[50] José MP Nascimento and José MB Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005.

[51] Javier Plaza, Antonio Plaza, Rosa Pérez, and Pablo Martinez. Joint linear/nonlinear spectral unmixing of hyperspectral image data. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 4037–4040. IEEE, 2007.

[52] BO Wu, Liang-pei Zhang, and Ping-xiang Li. Unmixing hyperspectral imagery based on support vector nonlinear approximating regression. *JOURNAL OF REMOTE SENSING-BEIJING-*, 10(3):312, 2006.

[53] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[54] Michael E Winter. N-findr: An algorithm for fast autonomous spectral endmember determination in hyperspectral data. In *Imaging Spectrometry V*, volume 3753, pages 266–276. International Society for Optics and Photonics, 1999.

[55] Marian-Daniel Iordache, José Bioucas-Dias, and António Plaza. Unmixing sparse hyperspectral mixtures. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 4, pages IV–85. IEEE, 2009.

[56] Daniel C Heinz et al. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE transactions on geoscience and remote sensing*, 39(3):529–545, 2001.

[57] Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[58] Chein-I Chang and Qian Du. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Transactions on geoscience and remote sensing*, 42(3):608–619, 2004.

[59] Chein-I Chang. A review of virtual dimensionality for hyperspectral imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2018.

[60] Giorgio Licciardi. Spectral unmixing. `https://www.sfpt.fr/hyperspectral/wp-content/uploads/2013/01/cours_Licciardi.pdf`, 2013. Course, GIPSA-lab, Grenoble-INP.

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

[62] Maurice D Craig. Minimum-volume transforms for remotely sensed data. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3):542–552, 1994.

[63] Antonio Plaza, Pablo Martínez, Rosa Pérez, and Javier Plaza. A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data. *IEEE transactions on geoscience and remote sensing*, 42(3):650–663, 2004.

[64] Michael E Winter. Comparison of approaches for determining end-members in hyperspectral data. In *Aerospace Conference Proceedings, 2000 IEEE*, volume 3, pages 305–313. IEEE, 2000.

[65] Chein-I Chang and Antonio Plaza. A fast iterative algorithm for implementation of pixel purity index. *IEEE Geoscience and Remote Sensing Letters*, 3(1):63–67, 2006.

[66] Antonio Plaza, David Valencia, and Javier Plaza. High-performance computing in remotely sensed hyperspectral imaging: The pixel purity index algorithm as a case study. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 8–pp. IEEE, 2006.

[67] Antonio Plaza, Javier Plaza, and Sergio Sánchez. Parallel implementation of endmember extraction algorithms using nvidia graphical processing units. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 5, pages V–208. IEEE, 2009.

[68] Xianyun Wu, Bormin Huang, Antonio Plaza, Yunsong Li, and Chengke Wu. Real-time implementation of the pixel purity index algorithm for endmember identification on gpus. *IEEE Geoscience and Remote Sensing Letters*, 11(5):955–959, 2014.

[69] Jinping Gu, Zebin Wu, Yonglong Li, Yufeng Chen, Zhihui Wei, and Wubin Wang. Parallel optimization of pixel purity index algorithm for hyperspectral unmixing based on spark. In *Advanced Cloud and Big Data, 2015 Third International Conference on*, pages 159–166. IEEE, 2015.

[70] P Stein. A note on the volume of a simplex. *The American Mathematical Monthly*, 73(3):299–301, 1966.

[71] Xuetao Tao, Bin Wang, Liming Zhang, and Jian Qiu Zhang. A new endmember extraction algorithm based on orthogonal bases of subspace formed by endmembers. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 2006–2009. IEEE, 2007.

[72] Ying Wang, Lei Guo, and Nan Liang. Using a new search strategy to improve the performance of n-findr algorithm for end-member determination. In *Image*

and Signal Processing, 2009. CISP'09. 2nd International Congress on, pages 1–4. IEEE, 2009.

[73] Wei Xiong, Chein-I Chang, Chao-Cheng Wu, Konstantinos Kalpakis, and Hsian Min Chen. Fast algorithms to implement n-findr for hyperspectral endmember extraction. *IEEE Journal of selected topics in applied earth observations and remote sensing*, 4(3):545–564, 2011.

[74] Fanxia Zeng, Maozhi Wang, Ke Guo, and Daming Wang. Initialization of the n-findr algorithm based on the max-min distance method. In *Intelligent Systems (GCIS), 2012 Third Global Congress on*, pages 378–381. IEEE, 2012.

[75] Joseph C Harsanyi and C-I Chang. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *IEEE Transactions on geoscience and remote sensing*, 32(4):779–785, 1994.

[76] Jun Li and José M Bioucas-Dias. Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, volume 3, pages III–250. IEEE, 2008.

[77] José M Bioucas-Dias. A variable splitting augmented lagrangian approach to linear spectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS'09. First Workshop on*, pages 1–4. IEEE, 2009.

[78] Tsung-Han Chan, Chong-Yung Chi, Yu-Min Huang, and Wing-Kin Ma. A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing. *IEEE Transactions on Signal Processing*, 57(11):4418–4432, 2009.

[79] ArulMurugan Ambikapathi, Tsung-Han Chan, Wing-Kin Ma, and Chong-Yung Chi. Chance-constrained robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4194–4209, 2011.

[80] Lidan Miao and Hairong Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777, 2007.

[81] Mark Berman, Harri Kiiveri, Ryan Lagerstrom, Andreas Ernst, Rob Dunne, and Jonathan F Huntington. Ice: A statistical approach to identifying endmembers in hyperspectral images. *IEEE transactions on Geoscience and Remote Sensing*, 42(10):2085–2095, 2004.

[82] Alina Zare and Paul Gader. Sparsity promoting iterated constrained endmember detection in hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(3):446–450, 2007.

[83] Nareenart Raksuntorn and Qian Du. A new linear mixture model for hyperspectral image analysis. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, volume 3, pages III–258. IEEE, 2008.

[84] Agustin Ifarraguerri and C-I Chang. Multispectral and hyperspectral image analysis with convex cones. *IEEE transactions on geoscience and remote sensing*, 37(2):756–770, 1999.

[85] Alina Zare and Paul Gader. Piece-wise convex spatial-spectral unmixing of hyperspectral imagery using possibilistic and fuzzy clustering. In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, pages 741–746. IEEE, 2011.

[86] José MP Nascimento and José M Bioucas-Dias. Learning dependent sources using mixtures of dirichlet: Applications on hyperspectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS'09. First Workshop on*, pages 1–5. IEEE, 2009.

[87] Liguo Wang, Danfeng Liu, and Qunming Wang. Geometric method of fully constrained least squares linear spectral mixture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 51(6):3558–3566, 2013.

[88] Hanye Pu, Wei Xia, Bin Wang, and Geng-Ming Jiang. A fully constrained linear spectral unmixing algorithm based on distance geometry. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2):1157–1176, 2014.

[89] ShihYu Chen, Yen-Chieh Ouyang, and Chein-I Chang. Recursive unsupervised fully constrained least squares methods. In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 3462–3465. IEEE, 2014.

[90] E Candès, T Tao, J Romberg, and R Baraniuk. Compressed sensing makes every pixel count. *International Journal on What's Happening In The Mathematical Sciences*, pages 114–127, 2007.

[91] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[92] David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2006.

[93] Jianying Sun, Qunbo Lv, and Jihao Yin. Applications of cs based spectrum recovery in hyperspectral images. In *Signal Processing (ICSP), 2014 12th International Conference on*, pages 928–933. IEEE, 2014.

[94] Lei Zhang, Wei Wei, Yanning Zhang, Fei Li, and Hangqi Yan. Structured sparse bayesian hyperspectral compressive sensing using spectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2014 6th Workshop on*, pages 1–4. IEEE, 2014.

[95] Matthew Fickus, Megan E Lewis, Dustin G Mixon, and Jesse Peterson. Compressive hyperspectral imaging for stellar spectroscopy. *IEEE Signal Processing Letters*, 22(11):1829–1833, 2015.

[96] Yuri Mejía, Henry Arguello, Facundo Costa, Jean-Yves Tourneret, and Hadj Batatia. Bayesian reconstruction of hyperspectral images by using compressed sensing measurements and a local structured prior. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 3116–3120. IEEE, 2017.

[97] Alfred M Bruckstein, David L Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM review*, 51(1):34–81, 2009.

[98] José M Bioucas-Dias and Mário AT Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, pages 1–4. IEEE, 2010.

[99] Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 45(3):600–616, 1997.

[100] José M Bioucas-Dias and Mario A. T. Figueiredo. Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5(2):354–379, 2012.

[101] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[102] José M Bioucas-Dias. A variable splitting augmented lagrangian approach to linear spectral unmixing. *First IEEE GRSS Workshop on Hyperspectral Image and Signal Processing-WHISPERS?2009, Grenoble, France*, 2009.

[103] Jonathan Eckstein and Dimitri P Bertsekas. On the douglas?rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1-3):293–318, 1992.

[104] Derek M Rogge, Benoit Rivard, Jinkai Zhang, and Jilu Feng. Iterative spectral unmixing for optimizing per-pixel endmember sets. *IEEE Transactions on Geoscience and Remote Sensing*, 44(12):3725–3736, 2006.

[105] José M Bioucas-Dias and Mário AT Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2992–3004, 2007.

[106] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Dictionary pruning in sparse unmixing of hyperspectral data. In *Hyperspectral Image and Signal Processing (WHISPERS), 2012 4th Workshop on*, pages 1–4. IEEE, 2012.

[107] Marian-Daniel Iordache, José M Bioucas-Dias, and Antonio Plaza. Total variation regulatization in sparse hyperspectral unmixing. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on*, pages 1–4. IEEE, 2011.

[108] José MP Nascimento, José M Bioucas-Dias, Jose M Rodriguez Alves, Vítor Silva, and Antonio Plaza. Parallel hyperspectral unmixing on gpus. *IEEE Geoscience and Remote Sensing Letters*, 11(3):666–670, 2014.

[109] Marian-Daniel Iordache, José M Bioucas-Dias, Antonio Plaza, and Ben Somers. Music-csr: Hyperspectral unmixing via multiple signal classification and collaborative sparse regression. *IEEE Transactions on Geoscience and Remote Sensing*, 52(7):4364–4382, 2014.

[110] José M Bioucas-Dias and José MP Nascimento. Hyperspectral subspace identification. *IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING*, 46(8):2435–2445, 2008.

[111] Joseph W Boardman. Automating spectral unmixing of aviris data using convex geometry concepts. 1993.

[112] Joseph W Boardman. Geometric mixture analysis of imaging spectrometry data. In *Geoscience and Remote Sensing Symposium, 1994. IGARSS'94. Surface and Atmospheric Remote Sensing: Technologies, Data Analysis and Interpretation., International*, volume 4, pages 2369–2371. IEEE, 1994.

[113] Manya V Afonso, José M Bioucas-Dias, and Mário AT Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE transactions on image processing*, 19(9):2345–2356, 2010.

[114] Marian-Daniel Iordache, Antonio Plaza, and José Bioucas-Dias. Recent developments in sparse hyperspectral unmixing. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 1281–1284. IEEE, 2010.

[115] Stephen Becker. Cosamp and omp for sparse recovery. `https://www.mathworks.com/matlabcentral/fileexchange/32402-cosamp-and-omp-for-sparse-recovery`, 2016.

[116] Deanna Needell and Joel A Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.

[117] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.

[118] José M Bioucas-Dias. Sunsal - matlab demo, under the 2010 section. `http://www.lx.it.pt/~bioucas/publications.html`, 2010.

[119] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ?1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.