UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria

Corso di laurea magistrale in Ingegneria dell'Automazione

Dipartimento di Ingegneria dell'Informazione

Tesi di laurea

Improvements to algorithms for hyperspectral linear unmixing based on statistical model

Laureanda:
Laura Dal Col

Relatori:
Ch.ma Prof.ssa **Maria Elena Valcher**
Università di Padova

Ch.mo Prof. **Gustavo Iván Marrero Callicó**
Universidad de Las Palmas de Gran Canaria

Correlatore:
Ch.mo Prof. **Sebastian López Suárez**
Universidad de Las Palmas de Gran Canaria

A.A. 2012-2013

"Read the directions and directly you will be directed in the right direction."
Alice:"I simply must get through!"
Doorknob: "Sorry, you're much too big. Simply impassible."
Alice:"You mean impossible?"
Doorknob: "No, impassible. Nothing's impossible."

Lewis Carroll, *Alice in Wonderland*

Abstract

The main contribution of this work is the development and implementation of new techniques for Hyperspectral Unmixing.

Spectral mixing is one of the main problems that arise when characterizing the spectral constituents residing at a sub-pixel level in a hyperspectral scene. It consists of the fact that many pixels in the scene are mixed in Nature, i.e., they are formed by different spectral constituents at sub-pixel levels. The main approach for dealing with the mixture problem in the literature has been Linear Spectral Unmixing, which assumes that the incident radiation interacts linearly with the material residing inside a pixel and, hence, the response obtained at the sensor can be modelled as a linear combination of a set of pure spectral signatures (called endmembers in hyperspectral unmixing jargon) weighted by their fractional coefficient in the pixel (called abundances in hyperspectral unmixing jargon). A descriptive model for the linear spectral mixture is the Normal Compositional Model, according to which the pixels of the hyperspectral image is a combination of random endmembers with known mean. Several techniques propose to estimate the mixture coefficients using a Bayesian algorithm coupled with a Markov Chain Monte Carlo sampling procedure. In this work we propose a improvement of these algorithms implementing a novel sampling approach inspired by Genetic Algorithms. Furthermore, many efforts have been made in order to reduce the computational complexity of the algorithm introducing linear approximation and parallelism for the task. These contributions to the Spectral Unmixing and, specifically, to the abundances estimation process can significantly improve the overall algorithm, allow for comparable performance with a complexity reduction. The simulations with synthetic data have been conducted using different noise levels in order to analyze the sensitivity of the proposed techniques. Experiments with real hyperspectral data have been conducted, using a image scene collected by the popular NASA's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS). All the experiments have been developed by using MATLAB Software.

Contents

Chapter 1

Hyperspectral Unmixing

Contents

1.1 The hyperspectral imaging concept

Hyperspectral imaging is an emerging and fast growing area of remote sensing. It is concerned with the measurement, analysis and interpretation of spectra acquired from a given scene (or specific object) at a short, medium or long distance by airbornes or satellite sensor.

To understand the advantages of hyperspectral imagery, we first review some basic spectral remote sensing concepts (see [Smith 2012], [Manolakis 2003] and [Peg Shippert 2003]). We recall that each photon of light has a wavelength determined by its energy level. Light and other forms of electromagnetic radiation are commonly described in terms of their wavelengths. Figure (1.1) shows the ranges of wavelength associated with different spectral bands.

Reflectance is the percentage of the light hitting a material that is then reflected by this one (as opposed to being absorbed or transmitted).

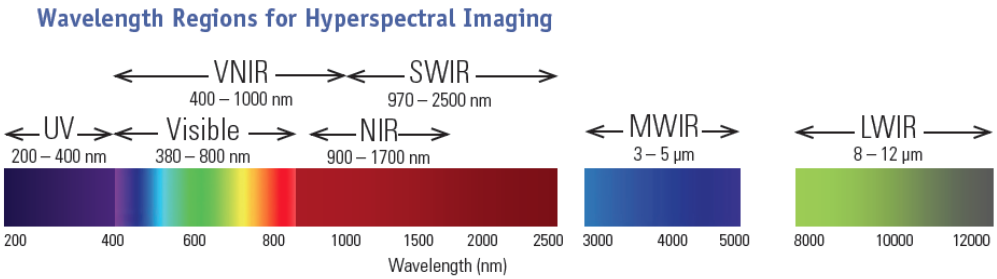


Figure 1.1: Wavelength bands: **UV** $(200 - 400\,nm)$, **VIS** $(380 - 800\,nm)$, **VNIR** $(400{-}1000\,nm)$, **NIR** $(900{-}1700\,nm)$, **SWIR** $(970{-}2500\,nm)$, **MWIR** $(3 - 5\,\mu m)$, **LWIR** $(8 - 12\,\mu m)$.

In remote sensing for hyperspectral data the fundamental property that we want to obtain is the *spectral reflectance*: the ratio of reflected energy to incident energy as a function of wavelength. Reflectance varies with wavelength for most materials because energy at certain wavelengths is scattered or absorbed in different degrees.

These reflectance variations are evident when we compare spectral reflectance curves (plots of reflectance versus wavelength) for different materials.

Field and laboratory spectrometers usually measure reflectance at many narrow, closely spaced wavelength bands, so that the resulting spectra appear to be continuous curves. Figure 1.2 illustrates an example of reflectance spectra of some common minerals acquired by multispectral remote sensors.

When a spectrometer is used in an imaging sensor, the resulting images record a reflectance spectrum for each pixel (see Figure 1.3).

The main characteristic of hyperspectral images is the high resolution they present in the spectral domain, since they are collected by instruments to measure hundreds of narrow spectral bands corresponding to continuous wavelength channel.

The very high spectral resolution of remotely sensed hyperspectral data, rooted in technological, modelling and processing advances, has fostered a strong interest in this image modality at an unprecedented rate in recent years. In fact, such a kind of data offers very significant potential in the identification of the materials present in a scene and their properties.

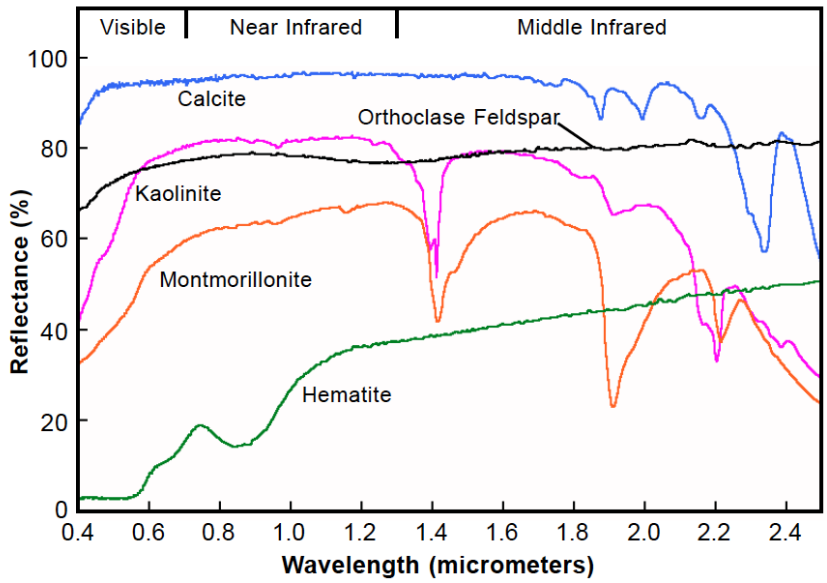


Figure 1.2: Reflectance spectrum reflectance for the mineral Calcite, Orthoclsae Feldspar, Kalonite, Montmoririllonite and Hematite with several absorption features.

However, the high dimensionality of hyperspectral data (and it's usually low spatial resolution) raises a series of new challenges on several fronts, namely, in signal and image processing, physical modeling, sensor design and calibration and computationally efficient processing.

The wealth of information avaliable from hyperspectral imaging instruments has opened a ground-breaking perspectives in several applications, including environmental modelling and assestments for earth-based and atmospheric studies, risk/hazard prevention and response, including wild land fire tracking, biological threat detection, monitoring of oil spills and other type of chemical contamination, target detection for military and defense/security purposes, una urban planning and manager study, among any others.

For instance, hyperspectral data generated from airborne sensors like the NASA's *Airborne Visible/Infrared Imaging Spectrometer* (AVIRIS) is now able to record the visible and near-infrared spectrum (wavelength region from 400 and 2500 nanometers) of the reflected light of an area from 2 to 12 kilometers wide and several kilometers long, using 224 spectral bands. Hyperspectral imaging sensors, provide image data containing both spatial and spectral in-

Acronim	Term
ANC	Abundance Nonnegativity Constraint
ASC	Abundance Sum-to-one Constraint
AVIRIS	Airborne Visible Infra-Red Imaging Spectrometer
ESA	Europe Space Agency
FCLS	Fully Constrained Least Square
SU	Spectral Unmixing
LMM	Linear Mixing Model
NASA	National Areonautic and Space Administration
NCLS	Nonnegative Constrained Least square
NCM	Normal Compositional Model
MSE	Mean Square Error
OPS	Orthogonal Subspace Projections
PCA	Principal Component Analysis
RMSE	Root Mean Square Error
SNR	Signal to Noise Ratio
VCA	Vertex Component Analysis

Table 1.1: List of acronyms.

formation, and this information can be used to address such detection tasks.

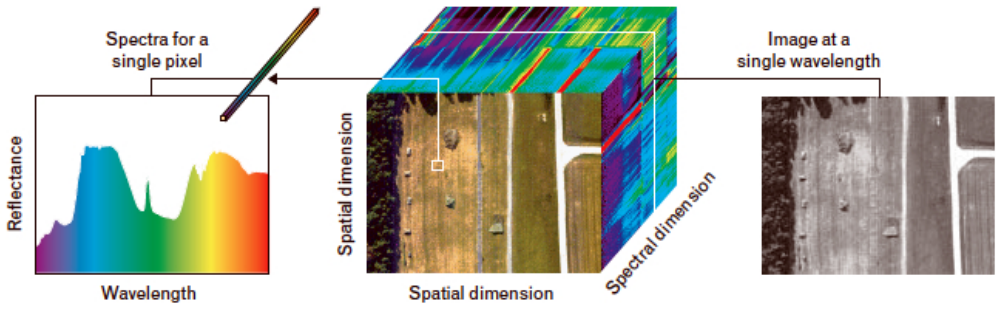


Figure 1.3: Basic data-cube structure (center) in hyperspectral imaging, illustrating the simultaneous spatial and spectral character of the data. The data cube can be visualized as a set of spectra (left), each for a single pixel, or as a stack of images (right), each for a single spectral channel.

	Hyperion	**Prisma**	**EnMAP**	**HyspIRI**
Country of origin	USA	Italy	Germany	USA
Spatial resolution	30 meters	5-30 meters	30 meters	60 meters
Revisit time	16 days	3/7 days	4 days	18 days
Spectral range	400-2500 nm	400-2500 nm	420-2450 nm	380-2500 nm
Spectral resolution	10 nm	10 nm	6.5-10 nm	10 nm
Swath width	7.7 km	30km	30 km	120 km
Earth coverage	Partial	Full	Full	Full
Launch	2000	2014	2014	2018
Lifetime	10 years	6 years	6 years	6 years

Table 1.2: Overview of some present and future remote sensing missions including hyperspectral sensors.

As a result, hyperspectral imaging sensors produce a three-dimensional data structure (with spatial-spectral components), referred to as a *data cube*. Figure 1.3 shows an example of such a data cube. In fact, each pixel can be considered as a high-dimensional vector where the values of a pixel comprise its associated spectral signature. The spectral signature is characteristic of each observed object and can be used as a *fingerprint* for identification purposes.

Although AVIRIS is a widely used platform, it constitutes only one source of hyperspectral data. Table 1.2 summarizes other international Earth observation missions with hyperspectral sensor already launched or to be launched in the near future.

The number and variety of processing tasks in hyperspectral remote sensing is enormous. However, the majority of algorithms can be organized according to the following specific tasks:

- **Dimensionality reduction** consists of reducing the dimensionality of the input hyperspectral scene in order to facilitate subsequent processing tasks.

- **Target and anomaly detection** consist of searching the pixels of a hyperspectral data cube for "rare" (either known or unknown) spectral signatures.

- **Change detection** consists of finding the "significant" (i.e. important to the user) changes between two hyperspectral scenes of the same geographic region.

- **Classification** consists of assigning a label (class) to each pixel of a hyperspectral data cube.
- **Spectral Unmixing** consists of estimating the fraction of the pixel area covered by each material present in the scene.
- **Compression** consists of reducing the size of the hyperspectral image without losing the relevant information that is needed in order to extract information from the scene.

In the following section we explore the problem of spectral unmixing and we summarize the most commonly used solutions available in the current literature in order to unmix hyperspectral images.

1.2 Spectral unmixing of hyperspectral data

The fact that the pixels of interest are frequently a combination of numerous disparate components has introduced a need to quantitatively decompose, or "unmix", these mixtures. Collecting data in hundreds of spectral bands, hyperspectral sensors have demonstrated the capability of performing spectral unmixing. In hyperspectral imagery, mixed pixels are mixtures of more than one distinct substance, and they exist for two reasons.

First, the spatial resolution of a sensor is low enough that disparate materials can jointly occupy a single pixel. The resulting spectral measurement will be a some composite of the individual spectra. This is the case of remote sensing platforms flying at a high altitude, where low spatial resolution is common.

Second, mixed pixel can result when distinct materials are combined into a intimate mixture. This circumstance can occur independent of the spatial resolution of the sensor.

Spectral unmixing is the procedure by which the measured spectrum of a mixed pixel is decomposed into a collection of constituent spectra, or *endmembers*, and a set of corresponding fraction, or abundances, that indicate the proportion of each endmember present in the mixed pixel. Endmembers normally correspond to familiar macroscopic objects in the scene, such as water, soil, vegetation, etc.

No matter the spatial resolution, the spectral signatures collected in natural environments are invariably a mixture of the signature of the various materials found within the spatial extent of the ground istantaneous field view of the imaging instruments.

For instance, the pixel vector labeled as "vegetation" in Figure 1.4 may actually comprise a mixture of vegetation and soil, or different types of soil and vegetation canopies. In this case, several spectrally pure signatures, i.e. the *endmembers*, are combined into the same (mixed) pixel.

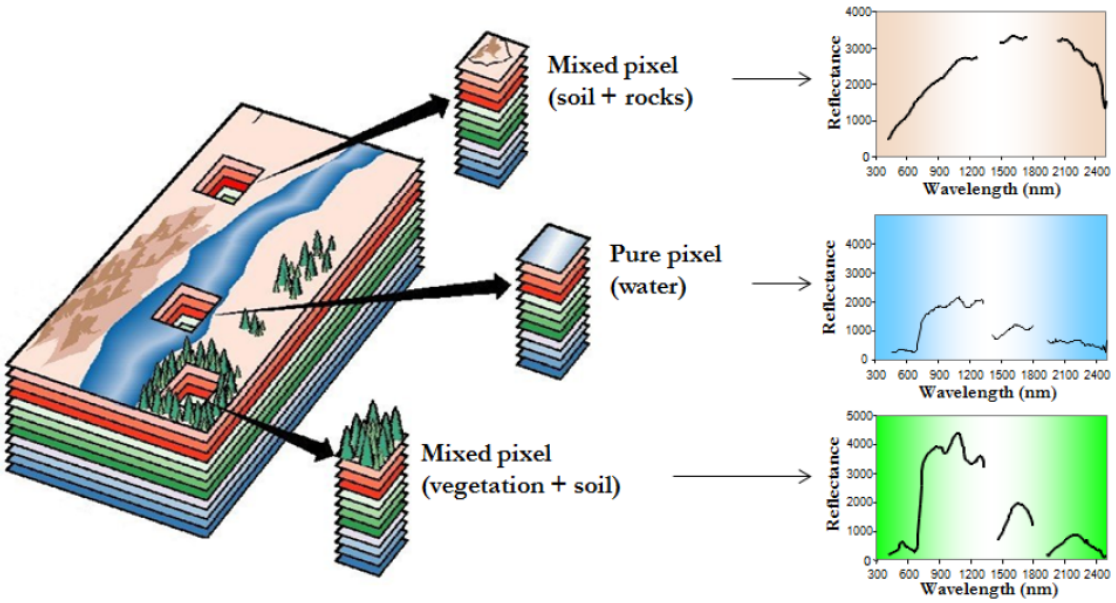


Figure 1.4: Mixed pixel in hyperspectral imaging.

There are a number of subtleties in this definition. First, the definition of pure material can be subjective and problem dependent. For example, suppose a hyperspectral image contains spectra measured from briks laid on the ground, the mortar between the bricks, and two types of plants that are growing through cracks in the brick. One may suppose that there are four endmembers. However, if the percentage of the area that is covered by the mortar is very small than we may not want to have an endmember of mortar, but only one *endmember* for "brick". If we have need to measure the mortar, then we may not care to distinguish between the plants since they may have similar signatures. On the other hand, suppose that one type of plant is desiderable and the other one is an invasive plant that needs to be removed. In this case we may want two plants endmembers. From this example is clear that the definition of endmembers can depend upon the application.

The availability of spectral images with a number of spectral bands that exceeds the number of spectral mixture components has allowed to cast the unmixing problem in terms of an over-determined system of equations in which, given a set of *endmembers*, the unmixing procedure can be defined in terms of a numerical inversion process.

Analytical models for the mixing of different material provide the foundation for developing techniques to recover estimates of the constituent substance spectra and their proportions for mixed pixel, i.e., the unmixing problem.

1.3 Linear and nonlinear mixing model

The basic premise of a mixture modeling is that within a given scene, the surface is dominated by a small number of distinct materials that have relatively constant spectral properties. These distinct substances (e.g., water, grass, mineral types) are called *endmembers* and the fraction in which they appear in a mixed pixel are called *fractional abundances*.

If most of the spectral variability within a scenes is a consequence of endmembers appearing in varying proportions, it logically follows that some combination of their spectral properties can model the spectral variability observed by the remote sensing system.

In Figure 1.5, the surface is portrayed as a *checkerboard* mixture, and any given package of incident radiation only interacts with one component (i.e., no multiple scattering between components). If the total surface area is considered to be divided proportionally according to the fractional abundances of the endmembers, then the reflected radiation will convey the characteristics of the associate media with the same proportions. In this case, the scattering and absorption of the incident electromagnetic radiation for any region in the surface, and thus the spectrum of a mixed pixel is a linear combination of the endmember spectra weighted by the fractional area coverage of each endmember in the pixel.

In some cases of interest the components of interest are in an *intimate* association (the endmember materials are mixed on spatial scales smaller than the path length of photons in the mixture), as illustrated in Figure 1.6. Let consider, for example, sand grains of different composition in a beach deposit: light typically interacts with more than one component as it is multipy scattered, and the mixing systematics between these different component are nonlinear.

Furthermore, the right-hand panel of Figure 1.6 illustrates a *multilayered* scene, where there are multiple interactions among the scatters at the different layers, that causes the nonlinearity of the mixing.

We can summarize the previous facts as follows:

- Linear mixing holds when the mixing scale is macroscopic and the inci-

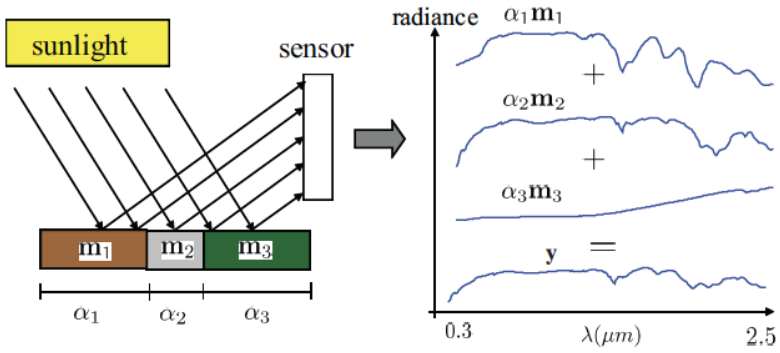


Figure 1.5: Linear mixing . The measured radiance at a pixel is a weighted average of the radiances of the materials present at the pixel.

dent light interacts with just one material, as it happens in checkerboard type scenes.

- Nonlinear mixing holds when the light suffers multiple scattering involving different materials.

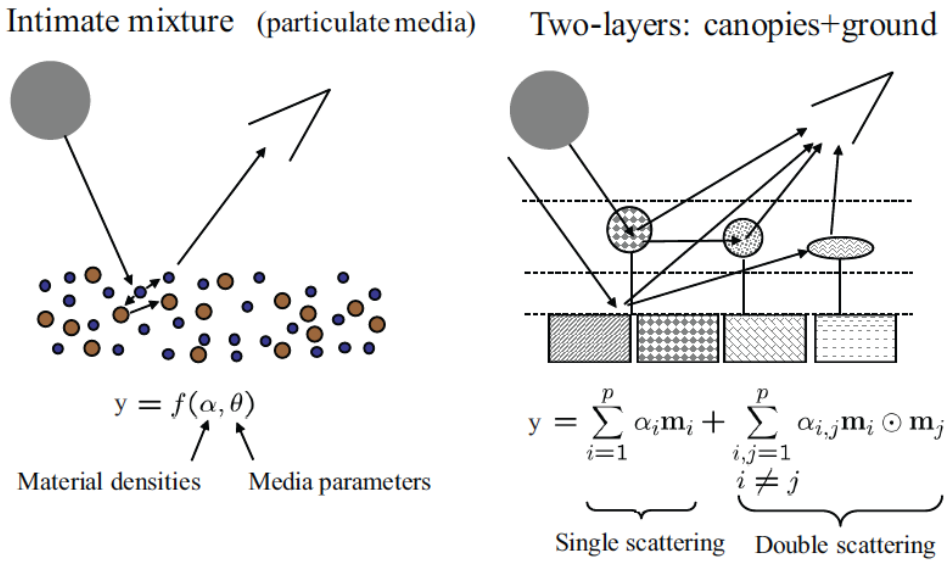


Figure 1.6: Two nonlinear mixing scenarios. Left hand: intimate mixture; Right hand: multilayered scene.

The question of whether the linear or nonlinear process dominate the spectral signatures of mixed pixels is still an unresolved issue. It likely depends on a number of factors and conditions in the scene. The linear approach has been

demonstrated in numerous applications to be a useful technique for interpreting the variability in remote sensing data, and a powerful means for converting spectral information into data products that can be related to the physical abundance of materials in the surface. Nevertheless, it is only strictly valid for the situation where the endmembers are arranged in discrete, segregated patches on the surface.

In the following, we focus on describing the linear spectral mixing scenario. Furthermore, all the presented unmixing algorithms relies on the hypothesis of linearity.

1.3.1 Linear Mixture Model

If the multiple scattering among distinct endmembers is negligible and the surface is partitioned according to the fractional abundances, as illustrated in Figure 1.5, then the spectrum of each pixel is well approximated by a linear mixture of endmember spectra weighted by the corresponding fractional abundances. In this case the definition of spectral measurement at the channel i, $i \in 1, \ldots, L$ (L is the total number of channels) from a given pixel, denoted by y_i, is given by the *Linear Mixing Model* (LMM):

$$y_i = \sum_{r=1}^{R} \rho_{ir} \alpha_r + w_i \tag{1.1}$$

where $\rho_{ir} \geq 0$ denotes the spectral measurement of endmember $r \in 1, \ldots, R$ at i^{th} spectral band, $\alpha_r \geq 0$ denotes the fractional abundance of endmember r , w_i denotes an additive perturbation (e.g., noise and modelling errors) and R denotes the number of endmembers.

For a given pixel, the fractional abundance α_r represents the fractional area occupied by the rth endmember. Therefore the fractional abundances are subject to the following constraints:

$$\text{Nonnegativity} : \alpha_r \geq 0, r = 1, \ldots, R \tag{1.2}$$

$$\text{Sum-to-one} : \sum_{r=1}^{R} \alpha_r = 1 \tag{1.3}$$

The fractional abundance vector $\boldsymbol{\alpha} \triangleq [\alpha_1 \ldots \alpha_R]^\top$ is in the standard $(R-1)$-simplex (or unit $(R-1)$- simplex). The nonnegativity and the sum-to-one constraints (1.2) and (1.3) are termed *abundance nonnegativity constraint* (ANC) and *abundance sum-to-one constraint* (ASC), respectively. We can sometimes expect that the abundance fractions sum to less than one since an

algorithm may not be able to account for every material in a pixel, for this reason the constraint can be relaxed or considered part of the modeling error.

Let $\boldsymbol{y} = [y_1 \dots y_L]^\top \in \mathbb{R}^L$ denote a L-dimensional column vector, and $\mathbf{m_r} = [\rho_{\mathbf{1r}} \dots \rho_{\mathbf{Lr}}]^\top$ denote the spectral signature of the rth endmember. Expression 1.1 can be rewritten in the compact form:

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\alpha} + \boldsymbol{w}$$

where $\boldsymbol{M} \triangleq [\boldsymbol{m}_1 \dots \boldsymbol{m}_R]$ is the mixing matrix containing the signatures of the endmembers present in the covered area, and $\mathrm{w} = [w_1 \dots w_L]^\top$.

Assuming that the columns of $\boldsymbol{M}$ are affinely independent, i.e., the set $\boldsymbol{m}_2 - \boldsymbol{m}_1, \boldsymbol{m}_3 - \boldsymbol{m}_2, \dots, \boldsymbol{m}_R - \boldsymbol{m}_1$ are linearly independent, then the set:

$$C \triangleq \{\boldsymbol{M}\boldsymbol{\alpha} : \sum_{r=1}^{R} \alpha_r = 1, \quad \alpha_r \geq 0, \quad r = 1, \dots, R\} \tag{1.4}$$

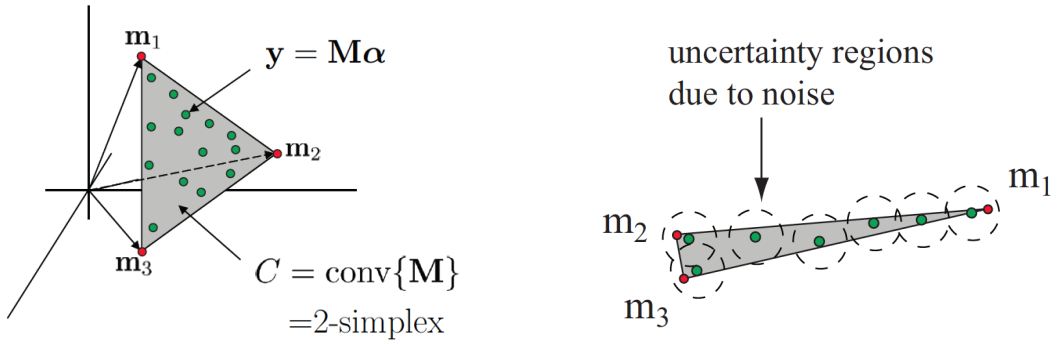
i.e. the convex hull of the columns of $\boldsymbol{M}$, is a $(R-1)$-simplex in $\mathbb{R}^L$.

The Linear Mixture Model can be interpreted in a graphic fashion by using a scatter plot between two bands or, more generally, between two no-colinear projection of the spectral vectors. For illustrative purposes, Figure 1.7-(a) provides a simple graphical interpretation in which the endmembers (points in red) are the most extreme pixels defining a simplex C which encloses all the other pixels in the data (points in green), so that we can express every pixel inside the simplex as a convex linear combination of the endmembers. As a results, a key aspect when considering the Linear Mixture Model is the correct identification of the endmembers, which are extreme points in the L-dimensional space.

1.4 Characterization of the spectral unmixing inverse problem

Given the data set $\boldsymbol{y} = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_N] \in \mathbb{R}^{L \times N}$ containing N L-dimensional spectral vectors, the spectral unmixing problem is, with reference to linear model (1.1), the estimation of the mixing matrix $\boldsymbol{M}$ and the abundances vectors $\boldsymbol{\alpha}_n = [\boldsymbol{\alpha}_{n1}, \dots, \boldsymbol{\alpha}_{nR}]^\top$ corresponding to pixels indexed by $n = 1, \dots, N$.

This is often a difficult inverse problem because the spectral signatures tend to be strongly correlated, yielding badly-conditioned mixing matrices and, thus, SU estimates can be highly sensitive to noise. This scenario is depicted in Figure 1.7-(b), where endmembers $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$ are very close,



(a) illustration of the simplex set C for $R = 3$ (C is the convex hull of the columns of $\boldsymbol{M}$, $C = \text{conv}(\boldsymbol{M})$).

(b) illustration of a badly-conditioned mixing matrix and noise (represented by uncertainty regions) centered on clean spectral vectors represented by green circles.

Figure 1.7: illustration of the concept of simplex.

thus yielding a badly-conditioned matrix $\boldsymbol{M}$, and the effect of noise is represented by uncertainty regions.

The solution of the linear spectral unmixing problem relies on two major requirements:

- A successful estimation of how many endmembers, R, are present in the input hyperspectral scene $\boldsymbol{y}$.
- The correct determination of a set $\boldsymbol{M} = \{\boldsymbol{m}_1, , \ldots, \boldsymbol{m}_R\}$ of R endmembers and their correspondent abundance fractions $\boldsymbol{\alpha}_n = [\boldsymbol{\alpha}_{n1}, \ldots, \boldsymbol{\alpha}_{nR}]^\top$ at each pixel $\boldsymbol{y}_n$, $n = 1, \ldots, N$.

In order to address these issues, a standard spectral unmixing chain consisting of three steps is generally applied. In a first step, a (optional) dimensionality reduction is conducted. This step is strongly connected with the estimation of the number of endmembers present in the hyperpectral scene. Once the number of endmembers has been determined, an endmembers extraction step identifies the spectral signatures present in a scene. Finally, the abundance estimation step produces as output the set of abundance maps associated to each extracted endmember.

Figure 1.8 shows the different steps involved in the processing chain, which are briefly summarized in the following sections.

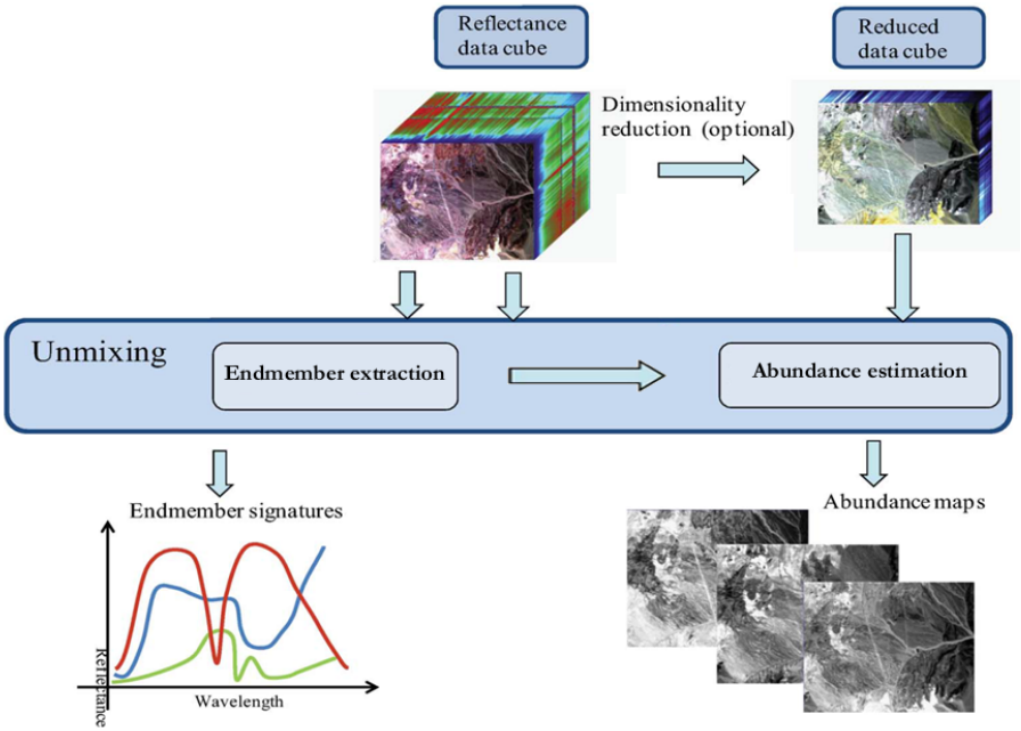


Figure 1.8: Spectral unmixing chain.

To characterize the linear spectral unmixing inverse problem, we use the *signal-to-noise-ratio* (SNR) :

$$SNR = \frac{\mathbb{E}[||\boldsymbol{x}||^2]}{\mathbb{E}[||\boldsymbol{w}||^2]} = \frac{trace(\boldsymbol{R_x})}{trace(\boldsymbol{R_w})}$$

where $\boldsymbol{R}_x$ and $\boldsymbol{R}_w$ are respectively, the signal (i.e., $\mathbf{x} = \mathbf{M}\alpha$) and noise correlation matrices and $\mathbb{E}$ denotes expected value.

1.4.1 Dimensionality reduction

The dimensionality of the space spanned by spectra from an image is often much lower than the available number of bands. Identifying appropriate subspaces facilitates dimensionality reduction, improving algorithm performance and complexity data storage. Furthermore, if the Linear Mixture Model is accurate, the signal subspace dimension is one less than equal to the number of endmembers, a relevant issue in hyperspectral unmixing.

1.4.2 Endmember extraction

This step consists of identifying the endmembers in the scene. *Geometrical* approaches exploit the fact that linearly mixed vectors are in a simplex set. *Statistical* approaches focus on using parameters estimation techniques to determine endmembers. Techniques may or may not include spatial information and assume (or not) the presence of pure pixels present in the data set.

On the last decade, several algorithms have been developed for automatic or semi-automatic extraction of spectral endmembers by assuming the presence of pure pixels in the hyperspectral data. Classic techniques include the PPI, N-FINDR, IEA, ORASIS, CCA, VCA, and ASP (see Table 1.1). One of the distinguishing properties of hyperspectral data is the multivariate information coupled with a two-dimensional (pictorial) representation amenable to image representation. Subsequently, most endmebers extraction algorithms listed above could benefit from an integrated framework in which both the spectral information and the spatial arrangement of pixel vectors are taken into account.

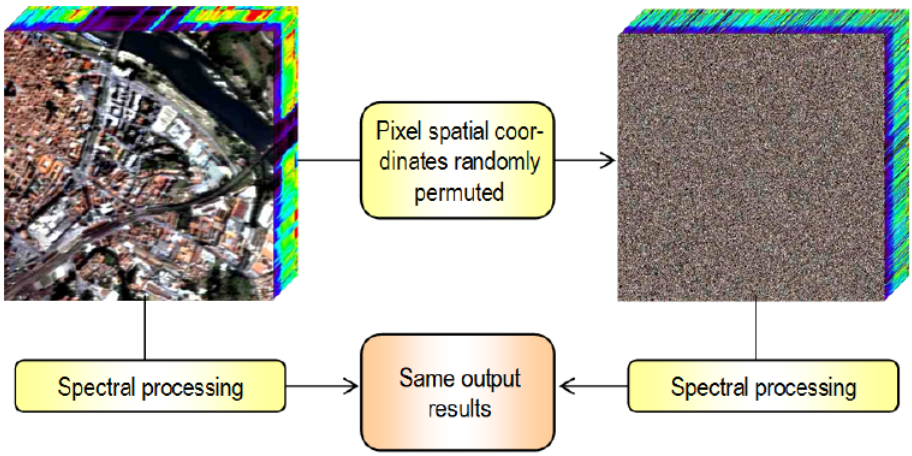


Figure 1.9: Example illustrating the importance of spatial information in hyperspectral analysis.

Figure 1.9 shows an example of this situation, in which a hyperspectral data cube collected over an urban area (high spatial correlation) is modified by randomly permuting the spatial coordiantes of the pixel vectors, thus removing spatial correlation. In both scenes, the application of a spectral-based endmember extraction method would yield the same analysis results, while it

is clear that a spatial-spectral technique could incorporate the spatial information present in the original scene into the endmember searching process. To the best of our knowledge, only a few attempts exist in literature aimed at including the spatial information in the process of extracting spectral endmembers.

In the following, we describe in more detail three selected spectral-based endmembers extraction algorithms. The reasons of this selection are:

- These algorithms are representative of the class of convex geometry-based and spatial processing-based techniques which have been successful in endmembers extraction.
- They are fully automated.
- They always produce the same final input results for the same input parameters.

1.4.2.1 N-FINDR

This algorithm (see [Winter 1999]) looks for the set of pixels with the largest possible volume by *inflating* a simplex inside the data. The procedure begins with a random selection of pixel (see Figure 1.10). Every pixel in the image must be evaluated in order to refine the estimate of endmembers, looking for the set of pixels that maximizes the volume of the simplex defined by selected endmembers. The volume of the simplex is calculated with every pixel in place of each endmember as follows.

Let $\mathcal{E} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_E\}$ denote the set of endmember vectors chosen to calculate the simplex volume (i.e., the set $\mathcal{E}$ is a subset of the columns of M), where $\boldsymbol{m}_i$ are column vectors of dimension $B \times 1$, for $i = 1, \ldots, E$.

Let $\boldsymbol{E}$ be defined as follows:

$$\boldsymbol{E} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \boldsymbol{m}_1 & \boldsymbol{m}_2 & \ldots & \boldsymbol{m}_E \end{bmatrix}$$

The volume $V(\mathcal{E})$ of the simplex formed by the endmembers is proportional to the determinant of $\boldsymbol{E}$, according to:

$$V(\mathcal{E}) = \frac{1}{(E-1)!}\text{abs}(|\boldsymbol{E}|)$$

In order to refine the initial volume estimate, a trial volume is calculated for every pixel in each endmember position by replacing that endmember and

recalculating the volume. If the replacement results in a volume increase, the pixel replaces the endmember. This procedure, which does not require any input parameters, is repeated until there are no replacements of endmembers left.

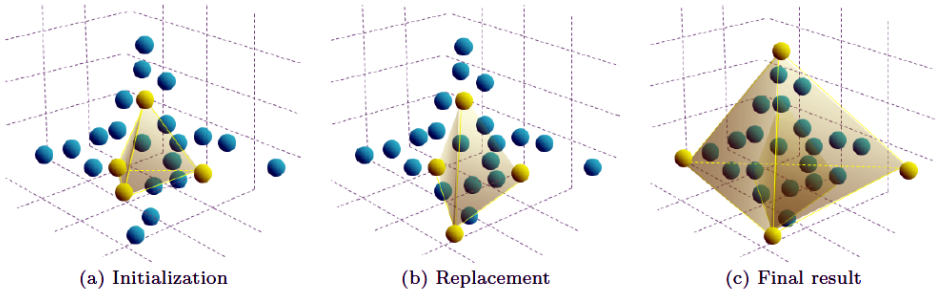


Figure 1.10: Graphical representation of N-FINDR algorithm.

1.4.2.2 Orthogonal Subspace Projection (OSP)

This algorithm (see [Harsanyi 1994]) starts by selecting the pixel vector with maximum length in the scene as the first endmember. Then, it looks for the pixel vector with the maximum absolute projection in the space orthogonal to the space linearly spanned by the initial pixel, and labels that pixel as the second endmember. A third endmember is found by applying an orthogonal subspace projector to the original image, where the signature that has the maximum orthogonal projection in the space orthogonal to the space linearly spanned by the first two endmembers. This procedure is repeated until the desired number of endmembers, R, is found. A shortcoming of this algorithm is its sensitivity to noise, since outliers are good candidates to be selected in the iterative process adopted by OSP. The VCA method discussed in the following subsection addresses this issue.

1.4.2.3 Vertex Component Analysis (VCA)

This algorithm ([Nascimento 2004]) also makes use of the concept of orthogonal subspace projections. However, as opposed to the OSP algorithm described above, the VCA exploits the fact that the endmembers are the vertices of a simplex, and that the affine transformation of a simplex is also a simplex.

As a result, VCA models the data using a positive cone, whose projection onto a properly chosen hyperplane is another simplex whose vertices are the

final endmembers. After projecting the data onto the selected hyperplane, the VCA projects all image pixels to a random direction and uses the pixel with the largest projection as the first endmember. The other endmembers are identified in sequence by iteratively projecting the data onto a direction orthogonal to the subspace spanned by the endmembers already determined, using a procedure that is quite similar to that used by the OSP. The new endmember is then selected as the pixel corresponding to the extreme projection, and the procedure is repeated until a set of R endmembers is found.

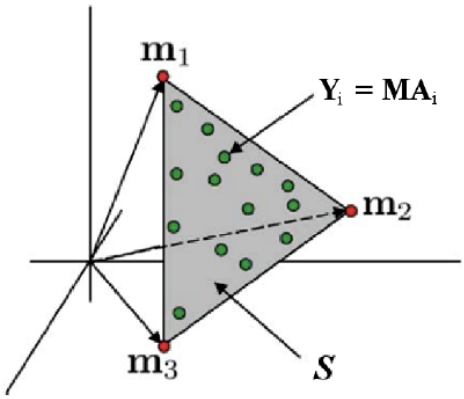


Figure 1.11: Graphical representation of VCA algorithm.

For illustrative purposes, Figure 1.11 shows a toy example depicting an image with three bands and three endmembers. Due to the mixing phenomenon all the data is in the plane S. If we project the data onto that plane we can represent the same data in two dimensions instead of three. Then we can apply the OSP to the projected dataset in order to obtain the endmembers. A possible shortcoming of the VCA algorithm can be illustrated by the following example: if there are two endmembers with similar spectral signatures and the power of the noise is high, then the subspace identification step could miss one of the two similar endmembers.

1.4.3 Abundance estimation

Given the extracted endmembers, the abundance estimation step consists of solving a constrained optimization problem which minimizes the residual between the observed spectral vectors and the linear space spanned by the inferred endmembers in order to derive fractional abundances which are, very often, constrained to be nonnegative and to sum-to-one (see equations (1.2) and(1.3)).

However, there are some hyperspectral unmixing approaches in which the endmembers determination and abundances estimation steps are implemented simultaneously.

Once a set of endmembers $\boldsymbol{M} = [\boldsymbol{m}_1 \dots \boldsymbol{m}_p]$ have been extracted, their correspondent abundance fractions $\boldsymbol{\alpha}$, corresponding to the pixel $\boldsymbol{y}$, can be estimated (in a least square sense).

The LSE is a mathematical technique that is used to implement an optimum estimation of parameters based on a certain known information. The estimation is regarded as optimum in the sense of minimizing the total energy of estimation errors. In the context of linear spectral unmixing, the LSE is implemented under an assumption of the LMM to obtain an optimum abundance estimation, given the information of mixed pixels and constituent pure pixels.

Using the LSE method, an optimum estimate $\hat{\boldsymbol{\alpha}}_{LS}$, of endmember abundances $\boldsymbol{\alpha}$ derived for the model (1.1), is given by the following unconstrained expression:

$$\hat{\boldsymbol{\alpha}}_{LS} \approx (\boldsymbol{M}^\top \boldsymbol{M})^{-1} \boldsymbol{M}^\top \boldsymbol{y} \tag{1.5}$$

However, it should be noted that the fractional abundances estimations obtained by means of (1.5) do not satisfy the ANC (1.2) and ASC (1.3) constraints. In this case a Nonnegative Constrained Least Squares (NCLS) algorithm can be used to obtain a solution to the ANC-constrained problem.

Let $\boldsymbol{A} \in \mathbb{R}^{R \times N}$ denote the column matrix containg the abundances of a set of N piexels. In order to take into consideration the ASC constraint, we replace the hard constraint $\mathbf{1}^\top \boldsymbol{A} = 1$ by the soft constraint $\sqrt{\delta}||\mathbf{1}^\top \boldsymbol{A} - 1||_2^2$ added to the quadratic term $||\boldsymbol{y} - \boldsymbol{M}\boldsymbol{\alpha}||_2^2$.

This is equivalent to use a new endmember signature matrix, denoted by $\boldsymbol{M'}$, and a modified version of the abundance estimates, denote by $\boldsymbol{\alpha}'$, are introduced as follows:

$$\boldsymbol{M}' = \begin{bmatrix} \boldsymbol{M} \\ \delta \mathbf{1}^\top \end{bmatrix} \quad \boldsymbol{\alpha}' = \begin{bmatrix} \boldsymbol{\alpha}' \\ \delta \mathbf{1} \end{bmatrix} \tag{1.6}$$

where $\mathbf{1} = [1 \dots 1]^\top \in \mathbb{R}^R$ and δ controls the the impact of thr ASC constraint. Using the expressions in (1.6), a fully constrained estimate can be directly obtained from the NCLS algorithm by replacing $\boldsymbol{M}$ and $\boldsymbol{\alpha}$ used in the NCLS algorithm with $\boldsymbol{M'}$ and $\boldsymbol{\alpha}'$.

Chapter 2

Bayesian Estimation of Linear Mixture using NCM

Contents

In the previous chapter we described the Linear Mixing Model (LMM) and also the most important techniques in the current literature for linear spectral unmixing.

However, in the last few years, new models attempt to describe the linear mixing scenario in a statistical fashion, using Bayesian analysis for this problem. Besides de well-known Linear Mixing Model (LMM), the Normal Compositional Model (NCM) is presented.

The objective of these models is to decompose a scene into a collection of spectra of pure materials, called *endmembers*, and a set of fractional *abundances* that indicate the proportion of each endmember in every pixel in the scene. This process, called spectral unmixing, can be divided in three main steps: the computation of the number of endmembers, the extraction of the spectral signatures of these endmembers and finally, the estimation of the corresponding abundances. The last step generally relies on the assumption of a linear mixing process, i.e., the mixed pixels are modeled as a convex combinations of endmembers.

The LMM assumes deterministic endmembers, which has led to some limitations when applied to real images [Keshava 2002].

Moreover, the endmembers extraction procedures based on the LMM can be inefficient when the extraction algorithms rely on the presence of pure pixels. This problem, outlined in [Nascimento 2004], is illustrated in Figure 2.1. This figure shows:

- The dual-band projections [on the two most discriminant axes identified by a principal component analysis (PCA)] of endmembers (red stars corresponding to the vertices of the red triangle).
- The dual-band domain containing all linear combinations of the endmembers (i.e., the red triangle).
- The dual-band simplex estimated by the N-FINDR algorithm using the black pixels. As there is no pixel close to the vertices of the red triangle, the N-FINDR estimates a much smaller simplex (in blue) than the actual one (in red).

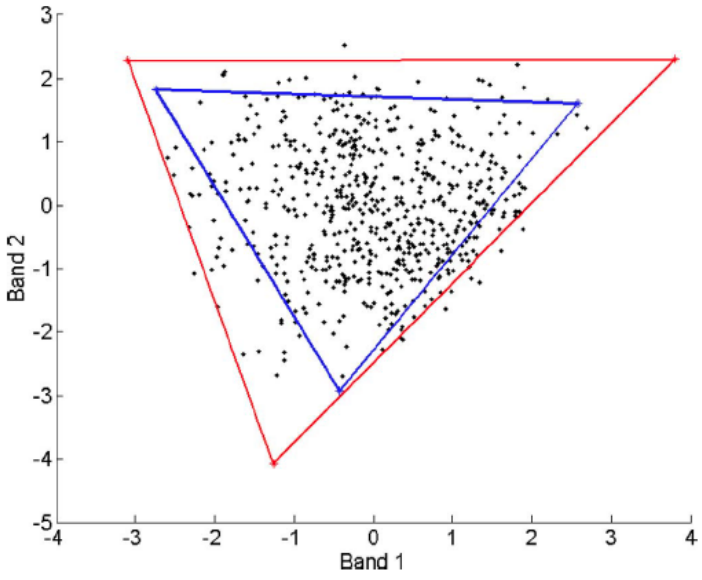
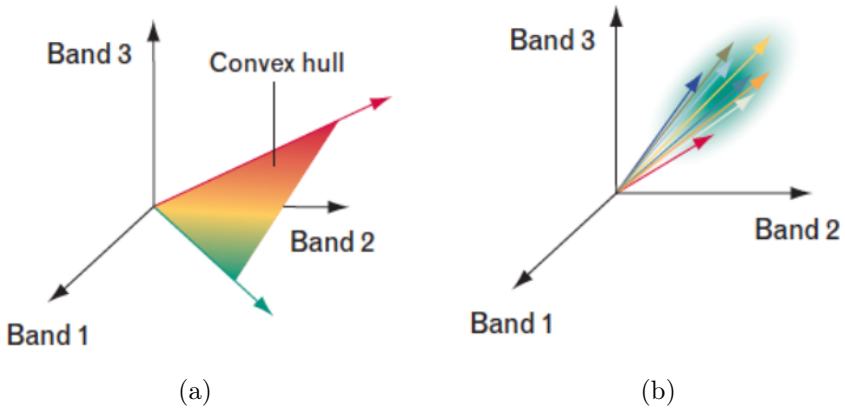


Figure 2.1: Scatterplot of dual-band correct (red) and incorrect (blue) results of the N-FINDR algorithm.

The NCM overcomes this issue by assuming *random* endmembers, which allows more flexibility regarding the observed pixels and the endmembers. In particular, the endmembers are allowed to be further from the observed pixels, which is clearly an interesting property for the problems in which spectral mixture are highly mixed [Dias 2012]. Figure 2.2 illustrates the

concept that lies behind the NCM: linear mixing models, with the aid of positivity and additivity constraints, permit spectra only in a simplex of the spectral space. Statistical models provide the probability for a spectrum to be at a certain region of the spectral space.



(a) (b)

Figure 2.2: Illustration of models for description of spectral variability in (a) Linear Mixing Model (b) Normal Compositional Model.

The main goal of this work is to analyze the spectral unmixing process based on NCM, in order to improve the algorithms related to the abundances estimation step.

In this chapter, the most important techniques in the current literature about spectral unmixing based on NCM are presented. In particular we focus our attention on the paper [Eches 2010], revising the statistical methods that will be used and thereby achieved results.

The chapter is organized as follows. First we introduce the NCM formalism in order to describe the linear mixture. Next we expose the Bayesian approach to NCM, defining the prior distribution of the parameters of the model. Then, we describe the Metropolis-Hastings sampling technique, that provides the tool for parameters estimation.

Finally, the procedure is tested on synthetic pixels, proving its effectiveness.

2.1 The Normal Compositional Model

The Normal Compositional Model assumes that the pixels of the hyperspectral image are linear combinations of random endmembers (as opposed to

deterministic for LMM) with known expected values (e.g., resulting from the endmembers extraction algorithms presented in Section 1.4.2) and unknow variance.

More formally, we assume that the spectrum of a mixed pixel can be written as follows:

$$\boldsymbol{y} = \sum_{r=1}^{R} \boldsymbol{\varepsilon}_r \alpha_r \tag{2.1}$$

where $\boldsymbol{\varepsilon}_r$ are independent gaussian vectors with known means $\boldsymbol{m}_r$, extracted from a spectral library or estimated by an appropriate method such as the ones presented in Section (1.4.2), and the same variance σ^2 in every spectral band, i.e. :

$$\boldsymbol{\varepsilon}_r | \boldsymbol{m}_r \sim \mathcal{N}(\boldsymbol{m}_r, \boldsymbol{I_L}\sigma^2)$$

where $\boldsymbol{I_L}$ is the $L \times L$ identity matrix, and $\mathcal{N}(\boldsymbol{m}, \boldsymbol{\Sigma})$ denotes the multivariate Gaussian distribution with mean vector $\boldsymbol{m}$ and variance $\boldsymbol{\Sigma}$.

The proposed model in (2.1) does not account for any possible correlation in the noise sequences but has been widely adopted in [Chang 1998] and [Li 2004].

However, the model in (2.1) can be easily modified to handle more complicated noise models with different variances in each spectral band as in [Veit], or by taking into account correlations between spectral bands as in [Ahmed 2013].

Note that there is no additive noise in (2.1) since the random nature of endmembers already models some kind of uncertainty regarding the endmembers.

The explicit expression of the probability density function associated with the endmember vectors $\boldsymbol{\varepsilon}_r$, $r = 1, \ldots, R$, given the parameter σ^2 and the mean vector $\boldsymbol{m}_r$ is:

$$f(\boldsymbol{x} = \boldsymbol{\varepsilon}_r | \sigma^2) = \frac{1}{[2\pi\sigma^2]^{L/2}} \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{m}_r||_2^2}{2\sigma^2} \right) \tag{2.2}$$

where $||\boldsymbol{x}||_2 = \sqrt{(\boldsymbol{x}^\top \boldsymbol{x})}$ is the l_2-norm.

Let suppose that the endmember spectra $\boldsymbol{\varepsilon}_1, \ldots \boldsymbol{\varepsilon}_n$ are *a priori* independent. The likelihood function of the observed pixel $\boldsymbol{y}$, given the abundance vector $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_R]^\top$ and the variance σ^2 values is :

$$f(\boldsymbol{y}|\,\boldsymbol{\alpha},\,\sigma^2) = \frac{1}{[2\pi\sigma^2 C(\boldsymbol{\alpha})]^{L/2}}\exp\left(-\frac{||\boldsymbol{y}-\boldsymbol{\mu}(\alpha)||^2}{2\sigma^2 C(\boldsymbol{\alpha})}\right) \tag{2.3}$$

That is a conditionally normal distribution with mean value and variance:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{y}|\boldsymbol{\alpha},\,\sigma^2] &= \mathbb{E}\left[\sum_{r=1}^{R}\boldsymbol{\varepsilon}_r\alpha_r|\boldsymbol{\alpha},\,\sigma^2\right] \\
&= \sum_{r=1}^{R}\alpha_r\mathbb{E}\left[\boldsymbol{\varepsilon}_r|\sigma^2\right] = \sum_{r=1}^{R}\alpha_r\boldsymbol{m}_r = \mu(\boldsymbol{\alpha}) \\
\mathbb{E}[(\boldsymbol{y}-\mu(\boldsymbol{\alpha}))^2|\boldsymbol{\alpha},\,\sigma^2] &= \mathbb{E}\left[\sum_{r=1}^{R}\alpha_r^2(\boldsymbol{\varepsilon}_r-\alpha_r)^2|\boldsymbol{\alpha},\,\sigma^2\right] \\
&= \sum_{r=1}^{R}\alpha_r^2\mathbb{E}\left[(\boldsymbol{\varepsilon}_r-\alpha_r)^2|\sigma^2\right] = \sigma^2\sum_{r=1}^{R}\alpha_r^2 = \sigma^2 C(\boldsymbol{\alpha})
\end{aligned}$$

where we define:

$$\mu(\boldsymbol{\alpha}) = \sum_{r=1}^{R}\alpha_j\boldsymbol{m}_r \quad C(\boldsymbol{\alpha}) = \sum_{r=1}^{R}\alpha_r^2 \tag{2.4}$$

In the following Section the Bayesian approach to NCM is presented, which provides a method for the abundances estimation under the assumption that the abundance vector $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_R]^\top$ and the variance parameter σ^2 are *a priori* independent.

2.2 Hierarchical Bayesian Model

The bayesian approach has the ability to model statistical variability and provides the inference machine for the estimation of the random parameters defined in the model.

The Bayesian viewpoint advocates that, just as the *a priori* knowledge about the model is contained in the *a priori* probability function, all the knowledge available about the unknown parameters is expressed in the *a posteriori* probability function.

The attribute *hierarchical* refers to the modelling strategy, involving additional degrees of dependece among the parameters model (called *hyperparameters*, defined by the respective *hyperpriors*).

Formally, given the model (2.1), let define the vector parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \sigma^2\}$ (where $\boldsymbol{\alpha}$ and σ^2 are assumed *a priori* independent), then the joint posterior distribution of $\boldsymbol{\theta}$ can be derived applying the Bayes paradigm:

$$f(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{f(\boldsymbol{y}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{f(\boldsymbol{y})} \tag{2.5}$$

In the equation (2.5), $f(\boldsymbol{y}|\boldsymbol{\theta})$ is the likelihood function (2.3) depending on the observation of the pixel $\boldsymbol{y}$ and $f(\boldsymbol{\theta})$ is the prior distribution which summarizes the prior knowledge regarding the unknown parameters. Since the observation $\boldsymbol{y}$ is assumed to be known, the function $f(\boldsymbol{y})$ is a normalization constant (also called *evidence*).

For the correct evaluation of the posterior distribution (2.5), it is crucial to choose an appropriate prior distribution for the NCM parameters, in order to satisfy the ANC (1.2) and ASC (1.3).

2.2.1 Parameters Priors

In this section we define the abundance prior in order to derive the posterior density function defined in (2.5).

Abundance Prior

Because of the sum-to-one constraint (1.3) inherent to the mixing model, the abundance vector can be rewritten as:

$$\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_{R-1} \,|\, \alpha_R]^\top = [{\boldsymbol{\alpha}^+}^\top \, \alpha_R]^\top$$

where $\boldsymbol{\alpha}^+ = [\alpha_1 \dots \alpha_{R-1}]^\top$ is the abundance sub-vector in which the last component is removed, and $\alpha_R = 1 - \sum_{r=1}^{R-1} \alpha_r$.

Moreover, to satisfy the ASC (1.3) the abundance sub-vector $\boldsymbol{\alpha}^+$ has to live in the simplex defined by:

$$\mathbb{S} = \left\{ \boldsymbol{\alpha}^+ \,|\quad \alpha_r \geq 0,\, \forall r = 1, \dots, R-1, \quad \sum_{r=1}^{R-1} \alpha_r \leq 1 \right\}$$

A uniform distribution on the simplex is chosen as prior distribution for the partial abundance vector $\boldsymbol{\alpha}^+$:

$$f(\boldsymbol{\alpha}^+) \propto \mathbf{1}_{\mathbb{S}}(\boldsymbol{\alpha}^+) \tag{2.6}$$

where $\mathbf{1}_{\mathbb{S}}(\cdot)$ is the indicator function of the set $\mathbb{S}$:

$$\mathbf{1}_{\mathbb{S}}(\boldsymbol{\alpha}^{+}) = \begin{cases} 1 & \text{if } \boldsymbol{\alpha}^{+} \in \mathbb{S} \\ 0 & \text{otherwise} \end{cases}$$

This prior distribution ensures the ASC (1.3) and ANC (1.2) of the abundance coefficients and reflects the absence of other prior knowledge regarding these parameters. Any abundance can be removed from $\boldsymbol{\alpha}$ and not only the last one α_R. For simmetry reason, any abundance coefficient can be removed from $\boldsymbol{\alpha}$, uniformly drawn in $\{1, \ldots, R\}$. For the sake of simplicity, this component is supposed to be α_R. Moreover, the notations $\mu(\boldsymbol{\alpha})$ and $C(\boldsymbol{\alpha})$ will be used in the sequel to denote the quantities in (2.4), where α_R is replaced by $1 - \sum_{r=1}^{R-1} \alpha_j$.

Endmember Variance Prior

Given the likelihood (2.2), assuming that the mean value $\boldsymbol{m}_r$ is fixed, and the variance is unkown, the prior distribution for the variance σ^2 is a conjugate inverse Gamma distribution[1] with hyperparameters ν and δ (See [Eches 2010]):

$$\sigma^2|\delta \sim \mathcal{IG}(x; \nu, \delta) = \frac{\delta^{\nu}}{\Gamma(\nu)} x^{-\nu-1} \exp\left(-\frac{\nu}{x}\right) \tag{2.7}$$

where $\Gamma(\nu) = \int_0^{\infty} t^{\nu-1} e^{-t} dt$.

The literature classically assumes $\nu = 1$ (in this case, $\Gamma(1) = 1$) and estimates δ using a hierarchical bayesian algorithm. In order to reflect the lack of knowledge regarding the hyperparameter prior, non-informative Jefferey's distribution[2] is is chosen as prior for δ:

$$f(\delta) \propto \frac{1}{\delta} \mathbf{1}_{\mathbb{R}^+}(\delta) \tag{2.8}$$

2.2.2 Posterior distribution of the parameters

The joint posterior distribution of the unknown parameter vector $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \sigma^2\}$ and the hyperparameter δ can be derived by means of the formula (2.5) using the hierarchical structure:

[1] Given a likelihood function, the **conjugate prior** is the prior distribution such that the prior and posterior are in the same family of distributions. The conjugate prior is often used in statistical inference in order to reduce the computation of the posterior, which can be derived in a closed form.

[2] this distribution preserve the uniformness under re-parametrization transform of random variabes.

$$f(\boldsymbol{\theta}, \delta|\boldsymbol{y}) \propto f(\boldsymbol{y}|\boldsymbol{\theta}) f(\boldsymbol{\theta}|\delta) f(\delta)$$

where the likelihood function $f(\boldsymbol{y}|\boldsymbol{\theta})$ and the hyperprior $f(\delta)$ have been defined in (2.3) and (2.8), respectively. Assuming the independence of the model parameters $\boldsymbol{\alpha}^2$ and σ^2 the prior distribution of $\boldsymbol{\theta}$ can be rewritten as:

$$f(\boldsymbol{\theta}|\delta) = f(\boldsymbol{\alpha}, \sigma^2|\delta) = f(\boldsymbol{\alpha}^+) f(\sigma^2|\delta)$$

Yielding the following expression for the joint posterior distribution of the parameters:

$$f(\boldsymbol{\theta}, \delta|\boldsymbol{y}) \propto \exp\left(-\frac{\|\boldsymbol{y} - \mu(\boldsymbol{\alpha})\|^2}{2\sigma^2 C(\boldsymbol{\alpha})} - \delta\right) \frac{1_{\mathbb{S}}(\boldsymbol{\alpha}^+) 1_{\mathbb{R}^+}(\delta)}{C(\boldsymbol{\alpha})^{L/2} \sigma^{L+2}} \tag{2.9}$$

The posterior distribution of (2.9) is too complex to derive the joint *maximum a posteriori* (MAP) estimator:

$$\hat{\boldsymbol{\theta}} = \{\hat{\boldsymbol{\alpha}}, \hat{\sigma}^2\} = \arg\max_{\boldsymbol{\alpha}, \sigma^2} f(\boldsymbol{\theta}, \delta|\boldsymbol{y})$$

In the context of Bayesian inference, Markov chain Monte Carlo (MCMC) methods are widely used. Such methods allow to simulate a distribution of interest by executing a random walk (Markov process) having a stationary density that is equivalent to the target (joint posterior) distribution (see [Gilks 1999]).

2.3 Hybrid Gibbs Sampler

The algorithms based on MCMC provide a way of reconstructing complete realizations of the process consistent with the data and the model. Moreover, the Gibbs sampling is used to obtain samples of the parameters to be estimated for each complete realization of the process. Conventional Gibbs sampling is achieved with the help of Bayes rule (2.5).

If the prior distribution $f(\boldsymbol{\theta})$ is chosen to be conjugate to the likelihood $f(\boldsymbol{y}|\boldsymbol{\theta})$ than the posterior can be calculated in a closed form. If the parameter vector is $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_k\} \in \mathbb{R}^k$, then the Gibbs sampler updates $\boldsymbol{\theta}$ component by component. The pseudocode of the algorithm is presented in Algorithm 1.

The algorithm can be combined with the Metropolis-Hastings sampler developed in [Dobigeon 2008] to produce a very flexible and powerful hybrid estimation algorithm. This version includes a reversible-jump step, which consents to accept or reject a proposed sample according to an acceptance

Algorithm 1 Gibbs Sampler

input: observed event $\mathbf{y}$.
output: samples $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_k\}$ of the unknow parameters vector $\boldsymbol{\theta}$.

1. Assign values to $\boldsymbol{\theta}^{(0)}$ drawn from the prior distribution $f(\boldsymbol{\theta})$.
2. Set $i = 0$.
3. repeat:
   - draw $\theta_1^{(i+1)}$ from $f(\theta_1|\boldsymbol{y}, \theta_2^{(i)}, \ldots, \theta_k^{(i)}) \propto f(\boldsymbol{y}|\theta_1, \theta_2^{(i)}, \ldots, \theta_k^{(i)}) f(\theta_1)$
   - draw $\theta_2^{(i+1)}$ from $f(\theta_2|\boldsymbol{y}, \theta_1^{(i)}, \theta_3^{(i+1)}, \ldots, \theta_k^{(i)})$.
   - $\vdots$
   - draw $\theta_k^{(i+1)}$ from $f(\theta_k|\boldsymbol{y}, \theta_1^{(i)}, \ldots, \theta_{k-1}^{(i)})$.
   - set $i = i + 1$.
   - store every m^{th} value of $\boldsymbol{\theta}$ after an initial burn-in period.

probability. The hybrid Metropolis-within-Gibbs sampler algorithm is summarized in Algorithm 2.

Theory guarantees that the hybrid procedure implements a Markov chain (indexed by i) which (asimptotically as the number of iterations approaches infinity) generates samples from the posterior distribution $f(\boldsymbol{\theta}|\boldsymbol{y})$.

Since we must draw the samples from an equilibrium distribution of the Markov chain, a key problem is knowing how long to let the Markov chain run before using the sample. In general, it is extremely hard to determine how many iterations of the algorithm are required to reasonably approximate the target distribution.

So far, in the works [Eches 2010], [Dobigeon 2008] and [Dobigeon 2009] any adaptative convergence criterion is adopted in order to check the Markov chain evolution. The mentioned papers propose an *ad hoc* approach to compute a reference estimate from a large number of iterations to ensure convergence of the sampler and good approximation with respect to the real distribution.

There is a number of convergence diagnostic available (see [Gilks 1999]) but, by far, the most common is the inspection of the chain output to check for trends in the variability of the samples. This strategy is used in this work to provide more robust results and to speed-up the sampling step.

Algorithm 2 Hybrid Gibbs Sampler

input: observed event $\mathbf{y}$.

output: samples $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_k\}$ of the unknow parameters vector $\boldsymbol{\theta}$.

1. Assign values to $\boldsymbol{\theta}^{(0)}$ drawn from the prior distribution $f(\boldsymbol{\theta})$.
2. Set $i = 0$.
3. Repeat :
   - draw $\tilde{\theta}_1$ from $f(\tilde{\theta}_1|\boldsymbol{y}, \theta_2^{(i)}, \ldots, \theta_k^{(i)}) \propto f(\boldsymbol{y}|\tilde{\theta}_1, \theta_2^{(i)}, \ldots, \theta_k^{(i)}) f(\tilde{\theta}_1)$
   - set $\theta_1^{(i+1)} = \tilde{\theta}_1$ with acceptance probability:

$$p_1^{(i)} = \min\left\{1, \frac{f(\tilde{\theta}_1|\boldsymbol{y}, \tilde{\theta}_1, \theta_2^{(i)}, \ldots, \theta_k^{(i)})}{f(\theta_1^{(i)})|\boldsymbol{y}, \theta_1^{(i)}, \theta_2^{(i)}, \ldots, \theta_k^{(i)})}\right\}$$

   - else $\theta_1^{(i+1)} = \theta_1^{(i)}$.

     $\vdots$
   - draw $\tilde{\theta}_k$ from $f(\tilde{\theta}_k|\boldsymbol{y}, \theta_1^{(i)}, \ldots, \theta_{k-1}^{(i)})$.
   - set $\theta_k^{(i+1)} = \tilde{\theta}_k$ with acceptance probability:

$$p_k^{(i)} = \min\left\{1, \frac{f(\tilde{\theta}_k|\boldsymbol{y}, \theta_1^{(i)}, \ldots, \theta_{k-1}^{(i)}, \tilde{\theta}_k)}{f(\theta_k^{(i)}|\boldsymbol{y}, \theta_1^{(i)}, \ldots, \theta_{k-1}^{(i)}, \theta_k^{(i)})}\right\}$$

   - set $i = i + 1$.
   - store every m^{th} value of $\boldsymbol{\theta}$ after an initial burn-in period.

2.3.1 Hybrid Gibbs Sampler for Hyperspectral unmixing

This Section studies the hybrid Metropolis-within-Gibbs algorithm, particularized for the hyperspectral unmixing problem. The purpose is to generate samples according to the posterior (2.9).
The sampler iteratively generates $\boldsymbol{\alpha}$ according to $f(\boldsymbol{\alpha}|\boldsymbol{y}, \sigma^2)$, σ^2 according to $f(\sigma^2|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2)$ and δ $f(\delta|\sigma^2)$, as detailed below. The overall hybrid Gibbs sampler algorithm for hyperspectral unmixing is summarized in Algorithm 3.

Algorithm 3 Hybrid Gibbs sampler for hyperspectral unmixing based on NCM

input: observed event $\mathbf{y}$.
output: samples $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_R\}$ of the (unknown) parameter of interest.

1. Assign values to $\delta^{(0)}$ drawn from the prior distribution $f(\delta)$ in (2.8).
2. Assign values to $\sigma^{2(0)}$ drawn from the prior distribution $f(\sigma^2|\delta)$ in (2.7).
3. Set $i = 0$.
4. repeat:
   - draw $\boldsymbol{\alpha}^{(i)}$ from the probability density function $f(\boldsymbol{\alpha}|\boldsymbol{y}, \sigma^2)$ in (2.11) using Metropolis-within-Gibbs step.
   - draw $\sigma^{2(i)}$ from the probability density function $f(\sigma^2|\boldsymbol{y}, \boldsymbol{\alpha}, \delta)$ in (2.12).
   - draw $\delta^{(i)}$ from the probability density function $f(\delta|\sigma^2)$ in (2.13).
   - set $i = i + 1$.
   - store every m^{th} value of $\boldsymbol{\alpha}$ after an initial burn-in period.

Generation according to $f(\boldsymbol{\alpha}|\boldsymbol{y}, \sigma^2)$ The Bayes' theorem yields the following expression of the conditional probability:

$$f(\boldsymbol{\alpha}|\boldsymbol{y}, \sigma^2) \propto f(\boldsymbol{y}|\boldsymbol{\theta}) f(\boldsymbol{\alpha}) \tag{2.10}$$

which easily leads to:

$$f(\boldsymbol{\alpha}|\, \boldsymbol{y},\, \sigma^2) = \frac{1}{[2\pi\sigma^2 C(\boldsymbol{\alpha})]^{L/2}} \exp\left(-\frac{||\boldsymbol{y} - \boldsymbol{\mu}(\alpha)||^2}{2\sigma^2 C(\boldsymbol{\alpha})} \right) \mathbf{1}_{\mathbb{S}}(\boldsymbol{\alpha}^+) \tag{2.11}$$

Note that the conditional distribution of $\boldsymbol{\alpha}$ is defined on the simplex $\mathbb{S}$. As a consequence the abundance vector $\boldsymbol{\alpha}^+$ satisfy the positivity and sum-to-one

constraints. The generation of $\boldsymbol{\alpha}$ can be achieved using a Metropolis-within-Gibbs algorithm.

Generation according to $f(\sigma^2|\boldsymbol{y}, \boldsymbol{\alpha}, \delta)$ The posterior distribution of the variance σ^2 conditional can be determined as follows:

$$f(\sigma^2|\boldsymbol{y}, \boldsymbol{\alpha}, \delta) \propto f(\boldsymbol{y}|\boldsymbol{\theta}) f(\sigma^2|\delta)$$

and hence:

$$f(x = \sigma^2|\boldsymbol{y}, \boldsymbol{\alpha}, \delta) \propto x^{-\left(\frac{L}{2}+1\right)-1} \exp\left(-\frac{\|\boldsymbol{y} - \mu(\boldsymbol{\alpha})\|^2}{2C(\alpha)x} + \frac{\delta}{x}\right) \tag{2.12}$$

which is also an inverse Gamma distribution, $\mathcal{IG}(x; \nu^*, \delta^*)$, with $\nu^* = \frac{L}{2} + 1$ and $\delta^* = \frac{\|\boldsymbol{y} - \mu(\boldsymbol{\alpha})\|^2}{2C(\alpha)} + \delta$, since the inverse Gamma distribution and exponential distribution are conjugate.

Generation according to $f(\delta|\sigma^2)$ The conditional distribution of δ is:

$$\delta|\sigma^2 \sim \mathcal{G}\left(1, \frac{1}{\sigma^2}\right) \tag{2.13}$$

where $\mathcal{G}(\alpha, \beta)$ is the Gamma distribution (see [Robert 2005], p. 581) with shape parameters α and β.

2.4 Simulation Results on synthetic pixel

To demonstrate the feasibility, practicality and potential of the exposed algorithm, we present numerical results from applying the scheme to synthetic datasets.

In Chapter 4, we provide results from a more realistic simulation, where the presented algorithm is applied to real hyperspectral data. Its performance, in terms of estimation error and computational cost, are compared with the algorithms presented in Chapter 3.

In this Section is illustrated the application of the abundances estimation algorithm by means of the hybrid Gibbs Sampler, with simulations results associated to synthetic data. In this simulation, we consider a set of $P = 3$ pixels obtained through the mixture of $R = 3$ different pure materials representative of a suburban scene: construction concrete, green grass and red

brick, extracted from the spectral libraries distributed with the ENVI package [RSI]. The actual abundance values are chosen to represent different mixture levels: *intermediate*, *unbalanced* and *balanced*, to test the effectiveness of the algorithm forefront different situations.

$$\begin{array}{lll} \text{Pixel 1} & \boldsymbol{\alpha}_1 = & [\ 0.5 \quad 0.3 \quad 0.2\ ]^\top \\ \text{Pixel 2} & \boldsymbol{\alpha}_2 = & [\ 0.1 \quad 0.1 \quad 0.8\ ]^\top \\ \text{Pixel 3} & \boldsymbol{\alpha}_3 = & [\ 1/3 \quad 1/3 \quad 1/3\ ]^\top \end{array}$$

The spectra of these endmembers are represented in Figure (2.3). The reflectances are observed in $L = 846$ spectral bands. The Signal-to-Noise Ratio (SNR) has been tuned to different levels, i.e. SNR = 10, 20, 40, 60. In order to express the SNR in terms of endmembers variance, we can rewrite the model (2.1) as follows:

$$y = \sum_{r=1}^{R} \alpha_r(\boldsymbol{m}_r + \boldsymbol{n}_r) = \underbrace{\sum_{r=1}^{R} \alpha_r \boldsymbol{m}_r}_{signal} + \underbrace{\sum_{r=1}^{R} \alpha_r \boldsymbol{n}_r}_{error} \tag{2.14}$$

where $\boldsymbol{n}_r \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I_L}\sigma^2)$ is an additive Gaussian noise and $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_r$ are the endmembers mean vectors.

The SNR is related to the endmembers variance σ^2 according to the following formula:

$$\text{SNR} = \frac{\mathbb{E}\left[\left\|\sum_{r=1}^{R} \alpha_r \boldsymbol{m}_r\right\|^2\right]}{\mathbb{E}\left[\left\|\sum_{r=1}^{R} \alpha_r \boldsymbol{n}_r\right\|^2\right]} = \frac{1}{\sigma^2} \frac{\left\|\sum_{r=1}^{R} \alpha_r \boldsymbol{m}_r\right\|^2}{L \sum_{r=1}^{R} \alpha_r^2}$$

The actual values of SNR and the corresponding endemembers variance are shown in Table 2.1, in which the $\boldsymbol{\sigma_i^2}$ denotes the endmembers variance of the Pixel i, corresponding to a given SNR level.

Figure (2.3) shows the spectral mixture corresponding to Pixel 1: the red line represents the expected value $\boldsymbol{y} = \alpha_1\boldsymbol{m}_1 + \alpha_2\boldsymbol{m}_2 + \alpha_3\boldsymbol{m}_3$. The pixel observation is corrupted by an additive Gaussian noise with variance $\sigma^2 = 2.66 \times 10^{-5}$, corresponding to a Signal-to-Noise Ratio value of SNR = 40 realization (the realization is depicted in blue line).

The observations are processed by the proposed algorithm that consists of NMC = 25000 iterations of the Gibbs sampler with Nbi = 5000 burn-in iterations. Then the MAP estimator of the unknown abundance vector $\boldsymbol{\alpha}$

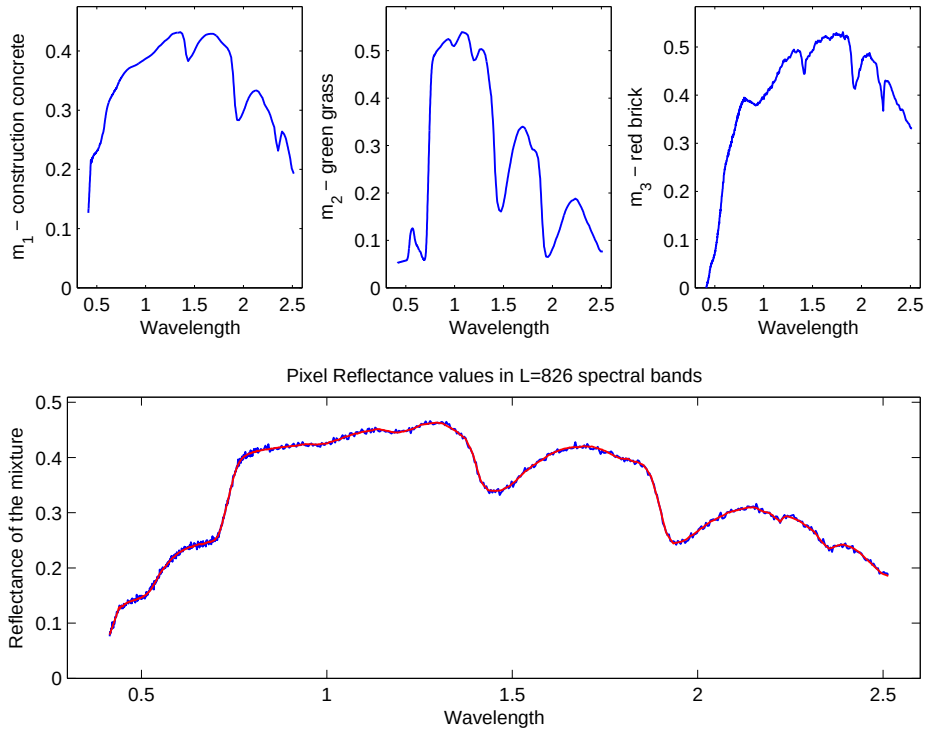


Figure 2.3: Spectral reflectance of the endmembers construction concrete, green grass and red brick (top) and reflectance of the mixed Pixel 1 (bottom) for noisy (blue line) and noisless (red line) case.

is computed by keeping the generated sample that maximizes the posterior distribution in (2.11):

$$\hat{\boldsymbol{\alpha}}_{MAP} \approx \arg\max_{\boldsymbol{\alpha}^+ \in \mathbb{S}} f(\boldsymbol{\alpha}|\, \boldsymbol{y},\, \sigma^2) \tag{2.15}$$

As the proposed algorithm generates samples distributed according to the posterior distribution in (2.11), these samples can be used to compute the posterior distributions of each parameter.

Figures (2.4), (2.5), (2.6) show the posterios $f(\alpha_i|\, \boldsymbol{y},\, \sigma^2)$ for $i = 1, 2, 3$ based on the observation of Pixel 1, Pixel 2 and Pixel 3 respectively, with different levels of SNR for each pixel. We can observe that the sampling procedure gives a more accurate reconstruction of the distribution (2.11) when the SNR takes high values. As consequence the MAP estimates based on the sampled posterior distributions are in good agreement with the real value of the abundances.

Table 2.1: Correspondence between endmembers variance and SNR used in the simulations.

SNR	σ_1^2	σ_2^2	σ_3^2
10 dB	0.026624	0.016455	0.03079
20 dB	0.0026624	0.0016455	0.003079
40 dB	2.6624 10e-5	1.6455 10e-5	3.079 10e-5
60 dB	2.6624 10e-7	1.6455 10e-7	3.079 10e-7

Table 2.2: MAP estimates of the endmembers variance corresponding to different SNR levels.

SNR	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\sigma}_3^2$
10 dB	0.026671	0.017804	0.026179
20 dB	0.0024944	0.0018139	0.0031896
40 dB	2.4619 10e-5	1.91153 10e-5	3.0192 10e-5
60 dB	2.5807 10e-7	1.8385 10e-7	3.1142 10e-7

In particular, we notice that the variance of the histograms in Figures (2.4), (2.5), (2.6) is higher when the SNR takes lower values, which means more uncertainty concearning these estimates.

Moreover, the same procedure can be used to compute the parameter σ^2, which corresponds to the endmembers variance in every spectral band. Figures (2.7), (2.8) and (2.9) shows the sampled posterior distribution of (2.12) by the proposed hybrid Gibbs algorithm. The sampled posterior distributions of the variance are clearly centered around the actual value. The MAP estimates $\hat{\sigma}_i^2$ of the variance σ_i^2 are shown in Table 2.2.

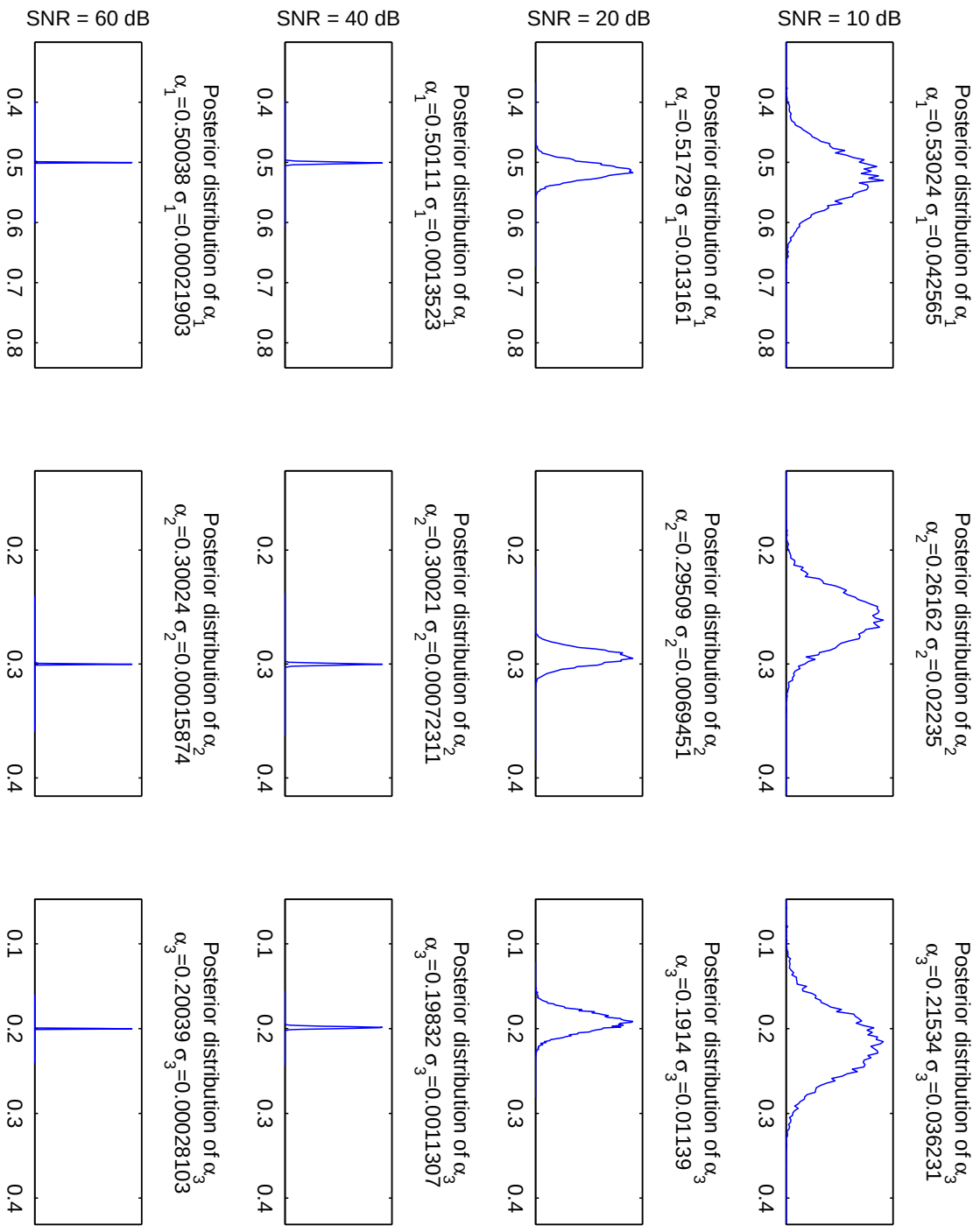


Figure 2.4: Posterior distribution of the abundance vector $\alpha_1 = [0.5 \quad 0.3 \quad 0.2]^\top$ of Pixel 1 and the MAP estimates for SNR = 10, 20, 40, 60 dB.

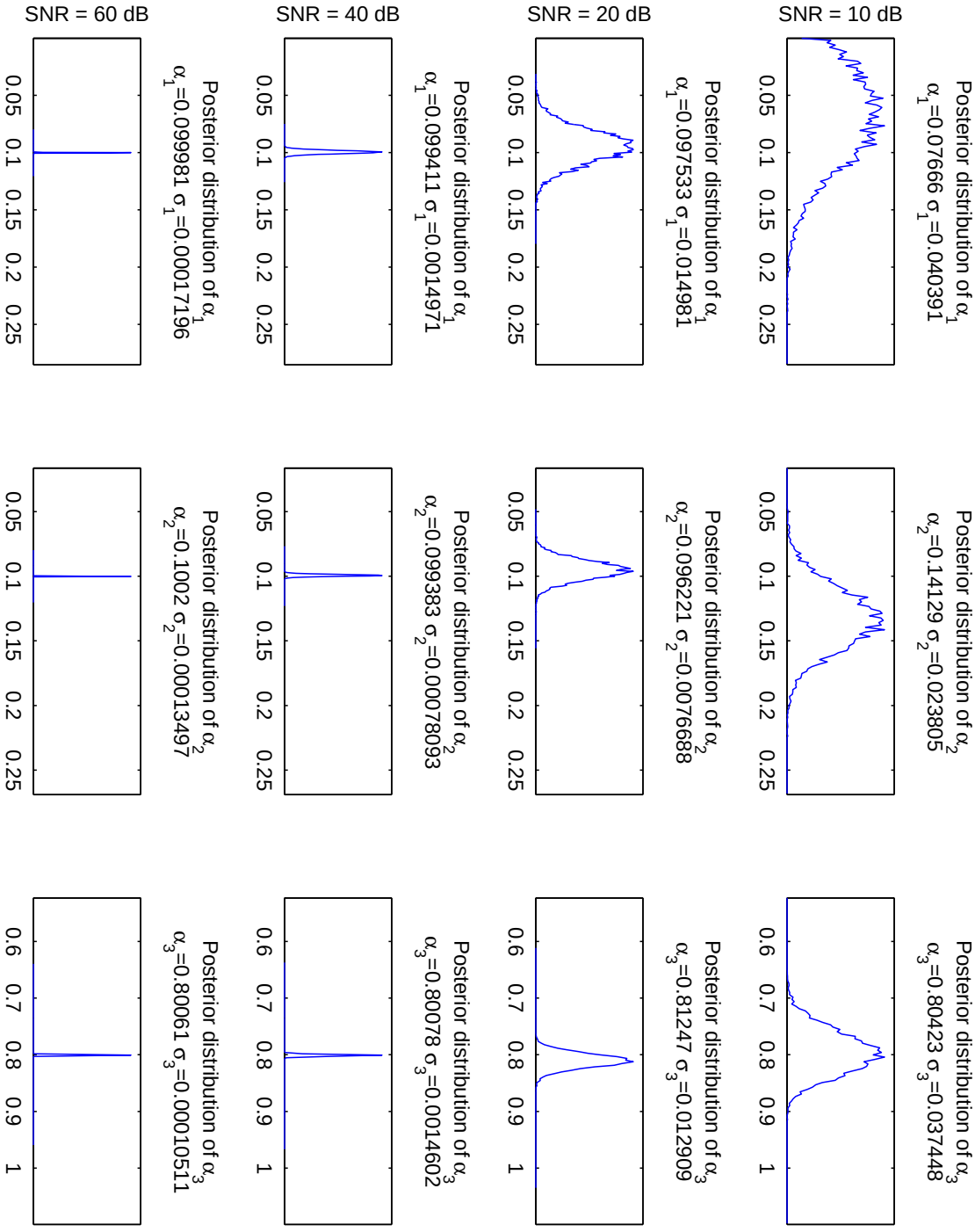


Figure 2.5: Posterior distribution of the abundance vector $\alpha_2 = [0.8 \quad 0.1 \quad 0.1]^\top$ of Pixel 1 and the MAP estimates for SNR = 10, 20, 40, 60 dB.

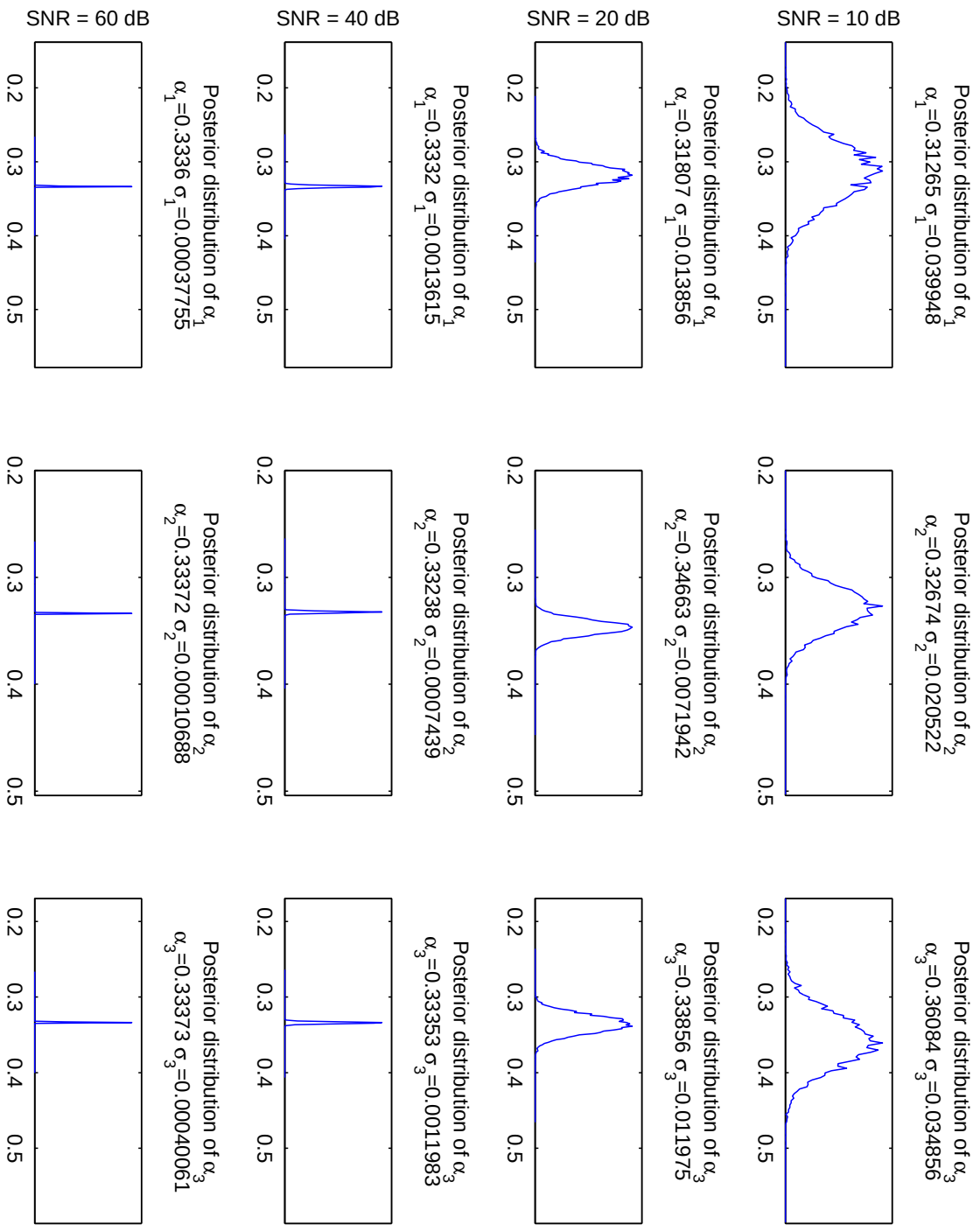


Figure 2.6: Posterior distribution of the abundance vector $\alpha_3 = [1/3 \quad 1/3 \quad 1/3]^\top$ of Pixel 1 and the MAP estimates for SNR = 10, 20, 40, 60 dB.

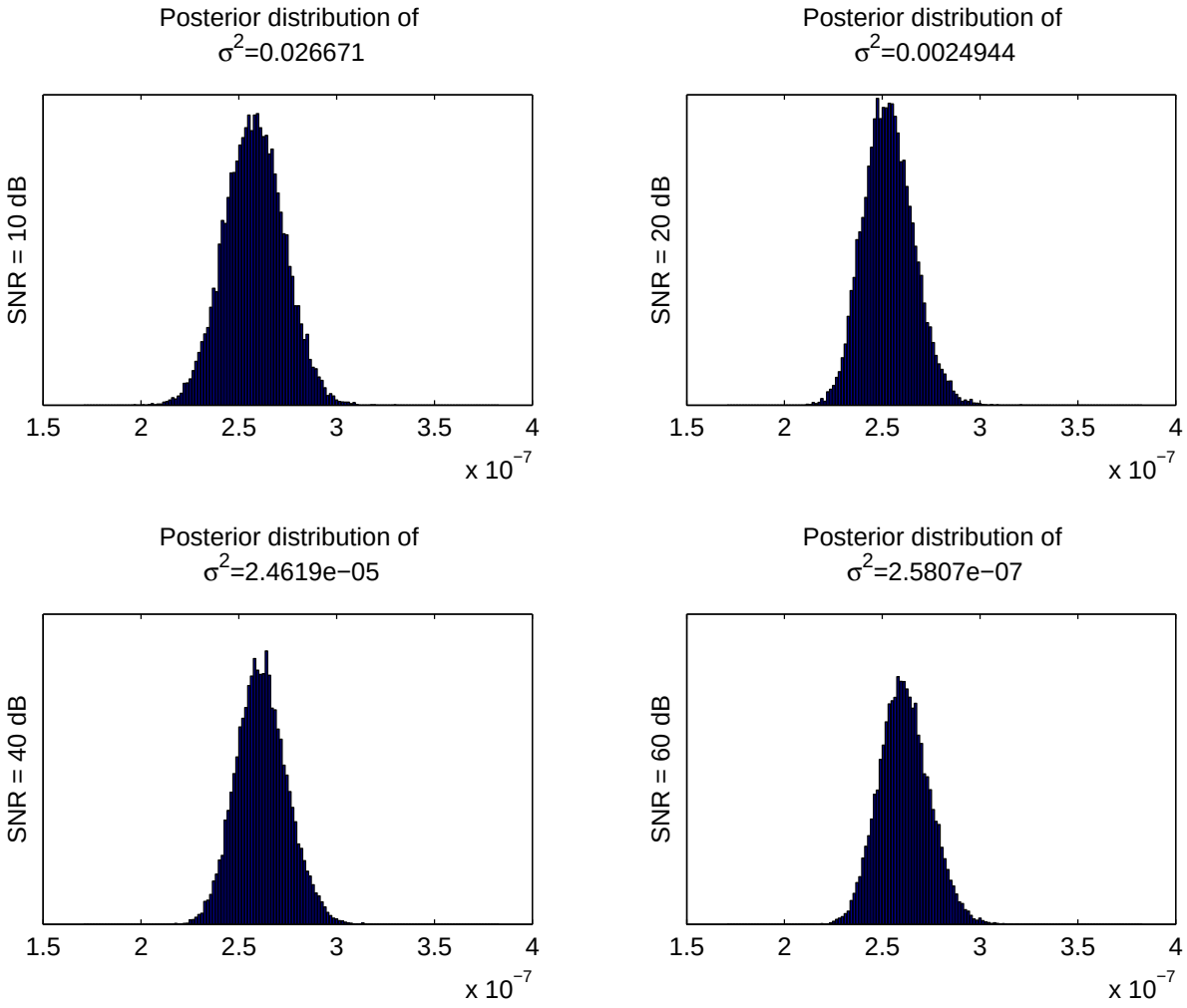


Figure 2.7: Posterior distribution of the variance σ_1^2 of Pixel 1 and the MAP estimates for SNR = 10, 20, 40, 60 dB.

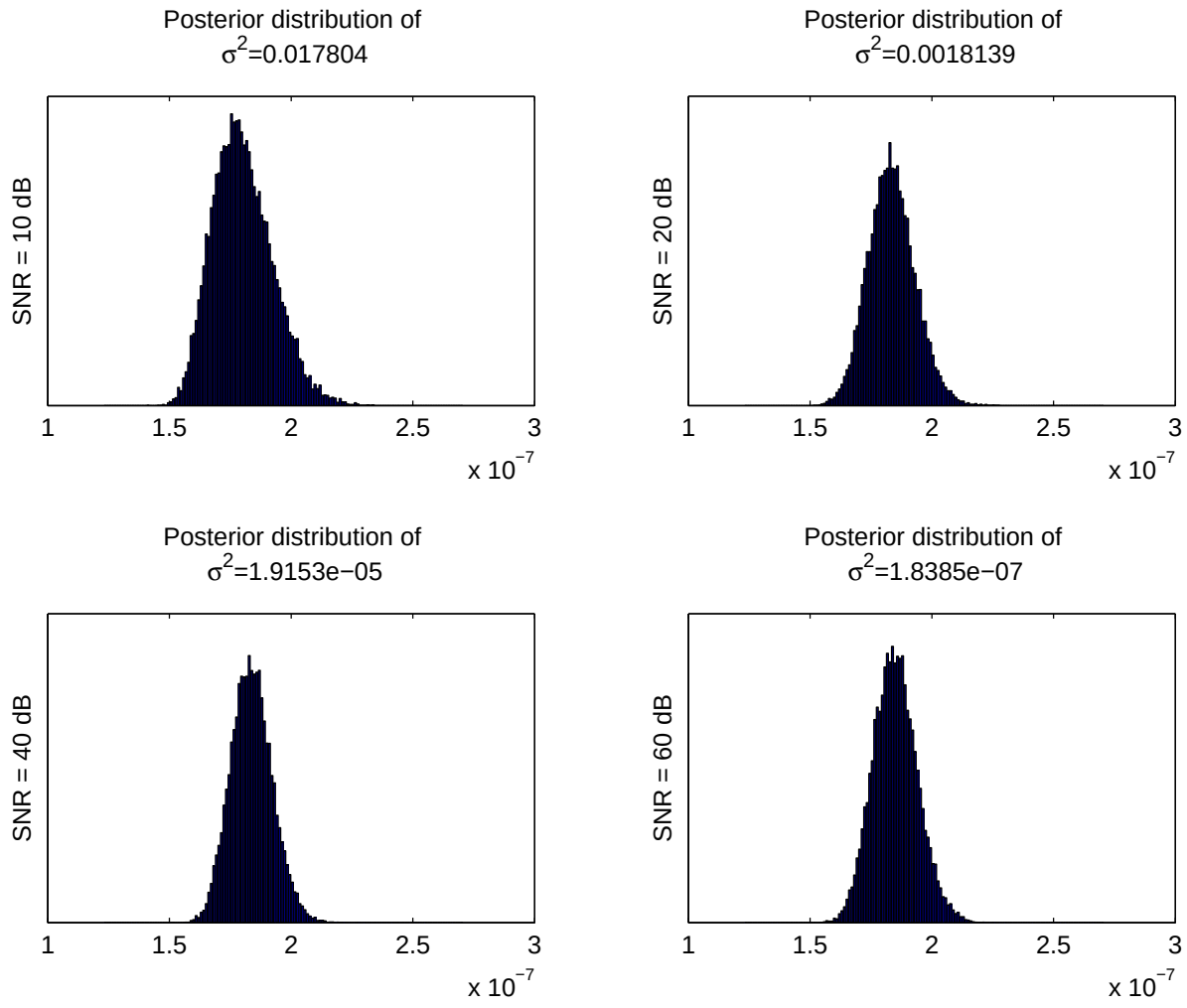


Figure 2.8: Posterior distribution of the variance σ_2^2 of Pixel 2 and the MAP estimates for SNR = 10, 20, 40, 60 dB.

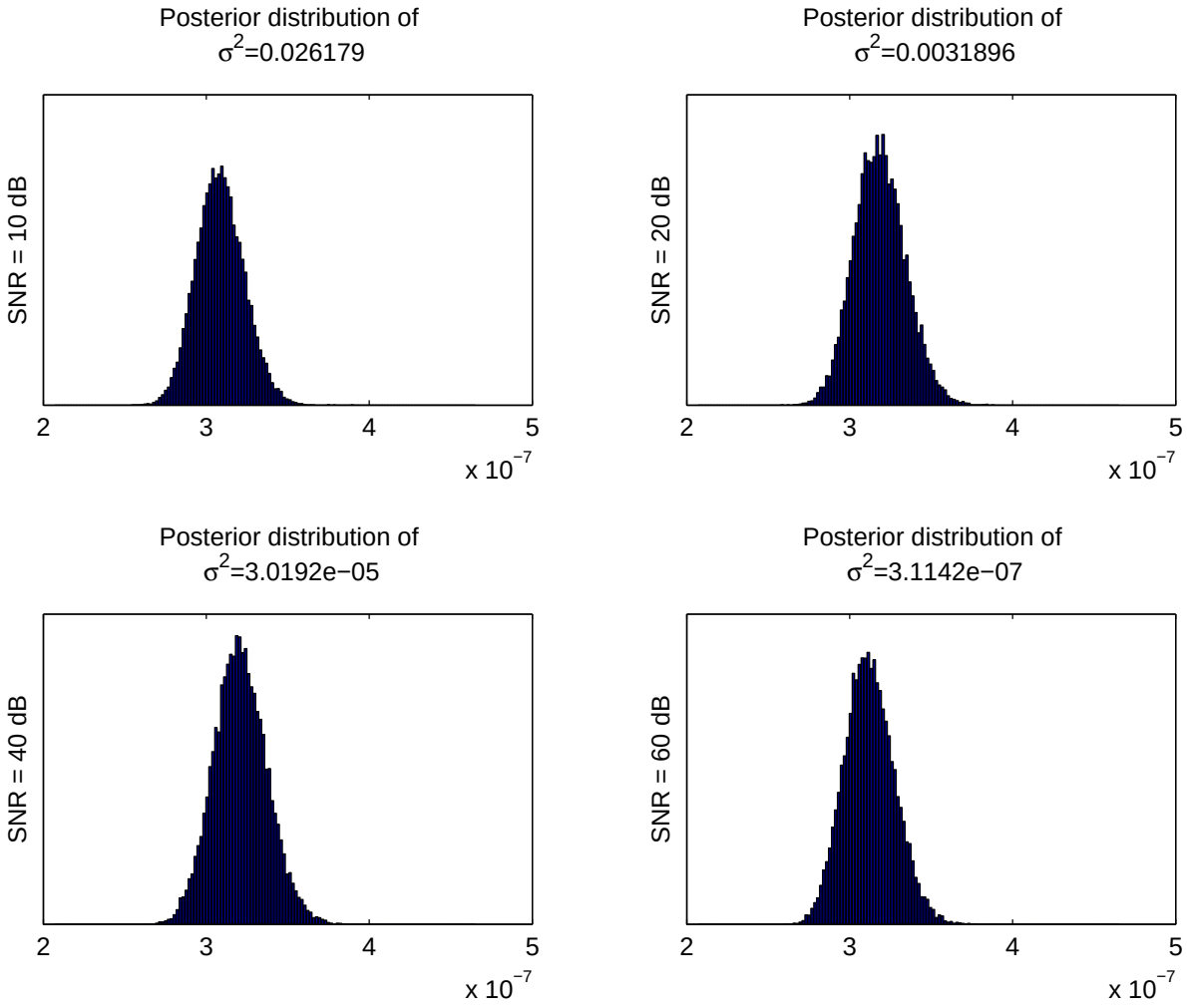


Figure 2.9: Posterior distribution of the variance σ_3^2 of Pixel 3 and the MAP estimates for SNR = 10, 20, 40, 60 dB.

Chapter 3

Algorithm improvements

Contents

In the previous chapter we described the unmixing procedure strategy based on NCM, using the hybrid Gibbs sampling techniques. Although the validity of the presented algorithm has been widely proved in [Eches 2010], no attempt has been conducted so far to improve its efficiency and to reduce his computational complexity.

Due to the extremely large volumes of data collected by imaging spectrometers and the need of real-time results (see [González 2013]), the improvement in terms of computational speed has received a considerable interest. In particular, it is important to achieve a high level of independent streams of operations in order to implement a parallel-processing system.

As mentioned before, the main objective of this work is to improve the algorithm presented in Chapter 2 using a novel technique in the hyperspectral field. For instance, one of the emerged problem is the lack of a convergence criterion regarding the sampled distribution (2.9) to the real distribution. In turn, we can achieve the selection of an appropriate paradigm to estimate the model parameters in (2.1), reducing the computational cost within an acceptable estimation error.

In this Chapter we illustrate our efforts towards complexity and computational reduction in the spectral unmixing technique based on NCM. The remainder of the Chapter is organized as follows. First, we introduce a convergence criterion for the Algorithm 3. Then we provide a more detailed description of the modification introduced in the sampling step of the algorithm [Eches 2010]. Then, we describe a novel framework to sample the posterior (2.9) on the parameters space $\mathbb{S}$ in an efficient way designing a step inspired to the Genetic Algorithm (GA). After that we explain our efforts towards linearization and parallelization of the algorithm structure. Finally we explain how to exploit spatial correlation in order to process an entire image. The theoretical and empirical assumptions that are exposed in the following exemplified using the data of Section 2.4.

3.1 Proposed convergence criterion

As mentioned in the previous Section, in the Gibbs sampling procedure for hyperspectral data (reported in Algorithm 3), it turns important, therefore, to determine the minimum number of samples required to ensure a reasonable approximation to the target posterior density.

The most common procedure, is a statistical approach to assessing convergence of the chain. With this approach, rather than considering the properties of the theoretical target distribution, only the statistical properties of the observed chain are analyzed. Reliance on the sample alone restricts such convergence criterion to heuristic. Nevertheless, experimental results over the hyperspectral data prove the effectiveness of this method.

Frequently, lack of convergence will be caused by poor mixing (Figure 3.1). Recall that *mixing* in MCMC context, refers to the degree to which the Markov chain explores the support of the posterior distribution. Poor mixing may stem from inappropriate proposals (if one is using the Metropolis-Hastings sampler, as in this case).

In this work a straightforward approach for assessing convergence is used, based on simply plotting and inspecting traces and histograms of the observed MCMC sample. If the trace of values for each of the stochastics exhibits asymptotic behavior, this may be a satisfactory evidence for convergence.

For that purpose, we add a diagnostic step at the end of each iteration to compute the progressive variances of the parameters posterior distributions. The overall algorithm is presented below (see Algorithm 4).

Algorithm 4 Hybrid Gibbs sampler for hyperspectral unmixing based on NCM with convergence diagnostic

input: observed event $\mathbf{y}$.
output: samples $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_R\}$ of the (unknown) parameter of interest.

1. Assign values to $\delta^{(0)}$ drawn from the prior distribution $f(\delta)$ in (2.8).
2. Assign values to $\sigma^{2(0)}$ drawn from the prior distribution $f(\sigma^2|\delta)$ in (2.7).
3. Set $i = 0$.
4. repeat:
   - draw $\boldsymbol{\alpha}^{(i)}$ from the probability density function $f(\boldsymbol{\alpha}|\boldsymbol{y}, \sigma^2)$ in (2.11) using Metropolis-within-Gibbs step.
   - draw $\sigma^{2(i)}$ from the probability density function $f(\sigma^2|\boldsymbol{y}, \boldsymbol{\alpha}, \delta)$ in (2.12).
   - draw $\delta^{(i)}$ from the probability density function $f(\delta|\sigma^2)$ in (2.13).
   - set $i = i + 1$.
   - store every value of $\boldsymbol{\alpha}$ after an initial burn-in period.
5. after an initial burn-in period compute the cumulative histogram of the sample for every set of N_k iterations.
6. Compute the histogram mean μ_H as reported in (3.1).
7. Compute the histogram variance σ_H^2 as reported in (3.2).
8. **if** the relative change in the variance from the previous iteration is less than a specified tolerance level, end.
9. **else** repeat.

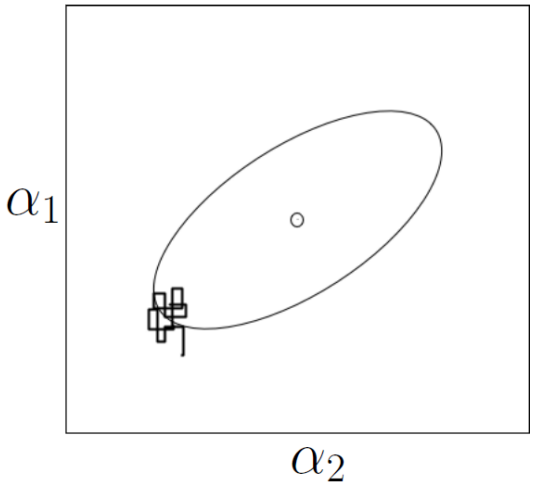


Figure 3.1: An example of a poorly-mixing sample in two dimensions. Notice that the chain is trapped in a region of low probability relative to the mean (dot) and variance (oval) of the true posterior quantity.

Let explain the convergence diagnostic more in detail. In the experiments, for every set of N_k iterations beyond some burn-in threshold N_b , we plot the histogram of each component α_i^j, $i = 1, \ldots, R$ of the collected samples:

$$\boldsymbol{\alpha}^j = \{\boldsymbol{\alpha}^{(N_b+1)}, \ldots, \boldsymbol{\alpha}^{(N_b+jN_k)}\} \quad j \in \mathbb{N}$$

The following graph represents the sample $\boldsymbol{\alpha}_j$, $j \in \mathbb{N}$ collected from the Markov Chain evolution.

Figure 3.2 shows the cumulative histograms obtained with $Nb = 50$ and $N_k = 50$ for the abundance vector of Pixel 1 (see Section 2.4) for $j = 4, 6$ (black and red line, respectively) using 200 and 300 samples. The blue line represents the final distribution obtained for a number of iteration that ensure convergency (the correspondent iteration numbers are reported in the title of each figure). We can notice that for a low SNR level (10 dB) the convergence is slow, due to the high level of noise present in the observations.

$$\boldsymbol{\alpha}^{(1)} \to \cdots \to \boldsymbol{\alpha}^{(N_b)} \to \overbrace{\overbrace{\boldsymbol{\alpha}^{(N_b+1)} \to \cdots \to \boldsymbol{\alpha}^{(N_b+N_k)}}^{\boldsymbol{\alpha}^1} \to \boldsymbol{\alpha}^{(N_b+N_k+1)} \to \cdots \to \boldsymbol{\alpha}^{(N_b+jN_k)}}^{\boldsymbol{\alpha}^j}$$

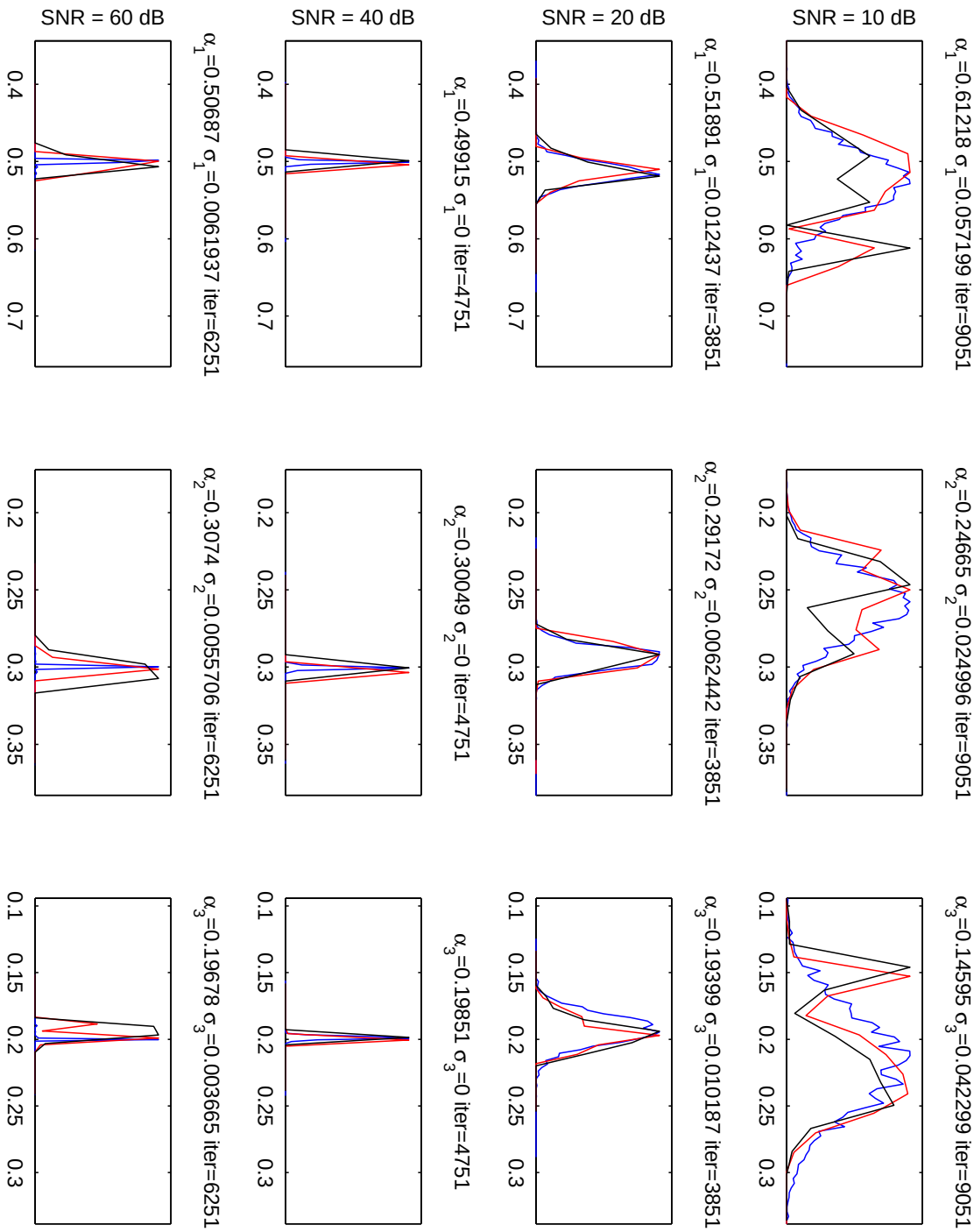


Figure 3.2: Cumulative histograms representing the Posterior distributions of the components of the abundance vector $\alpha_1 = [0.5 \quad 0.3 \quad 0.2]^\top$ of Pixel 1 reconstructed from the sample $\boldsymbol{\alpha}^j$, $j = 4$ (black line) and $j = 6$ (red line).

In this case we set the optimal number of classes for the histogram according to the *square-root* choice (see [Scott 1979]), which takes the square root of the number of data points in the sample. For each set of iterations, indexed by j, we have a data set with $N = jN_k$ points arranged in a frequency distribution of $N_c = \sqrt{N}$ classes, where N_c is an integer value.

Let x_i denote the representative (or class mark) of the i-th class, and f_i the frequency of the i-th class $i = 1, \ldots, N_c$. Then we can define the mean value and the variance associated with the j-th histogram as follows.

Histogram Mean Value The mean of a data set is the arithmetic average of the values in the set. Recall that when we summarize a data set in a frequency distribution, we are approximating the data set by *rounding* each value in a given class to the class mark. With this in mind, it is natural to define the mean of a frequency distribution by:

$$\mu_H = \frac{1}{N_c} \sum_{i=1}^{N_c} f_i x_i \tag{3.1}$$

The mean is a measure of the center of the distribution. As we can see from the algebraic formula, the mean is a weighted average of the class marks, with the relative frequencies as the weight factors.

Histogram Variance and Standard Deviation The variance of a data set is the arithmetic average of the squared differences between the values and the mean. When we summarize a data set in a frequency distribution, we are approximating the data set by *rounding* each value in a given class to the class mark. Thus, the variance of a frequency distribution is given by:

$$\sigma_H^2 = \frac{1}{N_c} \sum_{i=1}^{N_c} f_i (x_i - \mu_H)^2 \tag{3.2}$$

With respect to the notation introduced above, we can formally define the following convergence criterion:

The sequence $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots$ converges to $\boldsymbol{\alpha}$ when this condition holds. Given a $\varepsilon > 0$ there exists index j such that, defined $\sigma_H^2(j)$ the statistical variance of the histogram obtained from the sample $\boldsymbol{\alpha}^j$, we have:

$$\left\| \sigma_H^2(j+1) - \sigma_H^2(j) \right\| < \varepsilon \tag{3.3}$$

If the condition (3.3) is satisfied, than the statistical variance σ_H^2 asmptotically tends to a constant value $\bar{\sigma}_H^2$ and we can approximate $\sigma_H^2 \approx \bar{\sigma}_H^2$ for samples above $\boldsymbol{\alpha}^j$.

The algorithm actually determines the minimum index j for which the inequality (3.3) holds.

3.2 Motivations

Monte Carlo algorithms often depend on Markov chains to sample from very large data sets. A key ingredient in the design of an efficient Markov chain is determining rigorous bounds on how quickly the chain converges to its stationary distribution.

In particular, for this specific problem of finding the maximum of the sampled distribution (i.e., the MAP estimates of the parameters) we can *optimize* the sampling procedure exploring only the support of the posterior distribution in which is more probable (according to certain criteria) to find the maximum. In this context, the random walk that implements the Markov Chain in Algorithm 3 can be improved for this task.

Whith the above motivations in mind, in the next Section we develop a a new sampling strategy, that will be incorporated to the original Algorithm. This sampling strategy consists of applying the concepts of the Genetic Algorithms (GAs) in the hyperspectral imaging scenario.

3.3 Genetic Algorithms

Genetic Algorithms (GAs) are adaptive heuristic search algorithms premised on the evolutionary ideas of natural selection and genetics. The basic concept of GAs is to simulate processes in natural system necessary for evolution.

They are based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of *natural selection* and *survival of the fittest*. By mimicking this process, genetic algorithms are able to *evolve* solutions to real world problems, if they have been suitably encoded. GAs represent an intelligent exploitation of a random search within a defined search space to solve a problem.

We can resume the GAs principle as follows. GAs work with a population of *individuals*, each representing a possible solution to a given problem. Each individual is assigned a *fitness score* according to how good a solution to the

Genetic Algorithms	Hyperspectral imaging
individual	abundance vector $\boldsymbol{\alpha}$
genome	set of index $\{i_1, \ldots, i_{R-1}\}$ representing $\boldsymbol{\alpha}$
gene	a index i_k, $k = 1, \ldots, R$
population	set of abundance vectors $\boldsymbol{\alpha}_i$
fitnness function	a posteriori distribution $f(\boldsymbol{\alpha}\vert\, \boldsymbol{y}, \sigma^2)$
search space	the variation range of $\boldsymbol{\alpha}$ (the set $\mathbb{S}$)
fittest individual	abundance vector estimate $\hat{\boldsymbol{\alpha}}$

Table 3.1: Correspondence between Genetic Algorithms and Hyperspectral Imaging lexicon.

problem it is. The highly-fit individuals are given opportunities to *reproduce*, by *cross breeding* with other individuals in the population. This produces new individuals as *offspring*, which share some features taken from each *parent*. The least fit members of the population are less likely to get selected for reproduction, and so *die out*.

A whole new population of possible solutions is thus produced by selecting the best individuals from the current *generation*, and mating them to produce a new set of individuals. This new generation contains a higher proportion of the characteristics possessed by the best members of the previous generation. In this way, over many generations, good characteristics are spread throughout the population. By favouring the mating of the more fit individuals, the most promising areas of the search space are explored. If the GA has been designed well, the population will converge to an optimal solution to the problem.

In the Table 3.1 we translate the meaningful vocabulary presented above in the context of hyperspectral imaging, and in particular in terms of the unmixing process based on statistical procedures.

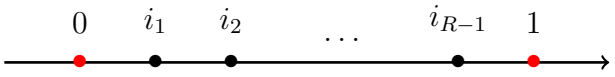


Figure 3.3: representation of each individual within a set of $R-1$ indices.

Before a GA can be run, a suitable coding (or representation) for the problem must be devised. In our case we decide to represent each individual

Algorithm 5 Genetic Algorithm for hyperspectral unmixing based on NCM (sampling of $\boldsymbol{\alpha}$)

input: the number n of individuals in a population and the fraction χ of the population to be replaced by crossover in each iteration.
output: the fittest individual $\hat{\boldsymbol{\alpha}}$

1. Assign random values to each individuals $\boldsymbol{\alpha}_i^{(0)}$ for $i = 1, \ldots, n$ of the population $\boldsymbol{P}^{(0)}$.
2. Evaluate the fitness of each individual $\boldsymbol{\alpha}_i^{(0)}$ for $i = 1, \ldots, n$ of the population $\boldsymbol{P}^{(0)}$.
3. set $k = 0$.
4. repeat:
   - Create generation $k + 1$
     (a) Copy: select $(\chi - 1)n$ members of $\boldsymbol{P}^{(k)}$ and insert into $\boldsymbol{P}^{(k+1)}$.
     (b) Selection, Crossover and Mutation: select $n\chi$ members of $\boldsymbol{P}^{(k)}$, pair them up, produce offspring, insert the offspring into $\boldsymbol{P}^{(k+1)}$
   - Evaluate the fitness of each individual $\boldsymbol{\alpha}_i^{(k+1)}$ for $i = 1, \ldots, n$ of the population $\boldsymbol{P}^{(k+1)}$.
   - Evaluate the fitness of each individual $\boldsymbol{\alpha}_i^{(k+1)}$ for $i = 1, \ldots, n$ of the population $\boldsymbol{P}^{(k+1)}$.
   - set $k = k + 1$.
5. end reapeat.

(i.e. the abundance vector $\boldsymbol{\alpha} \in \mathbb{R}^R$) through $R-1$ linearly independent indices $i_1, \ldots, i_{R-1}$ as depicted in Figure 3.3. The indices are numbers in the interval $[0, 1]$ arranged in increasing order, such that:

$$0 \leq i_1 \leq i_2 \leq \ldots i_{R-1} \leq 1$$

The abundance vector components $\alpha_1, \ldots \alpha_R$ are related to de indices i_k, with $k = 1, \ldots, R-1$ through the following formula:

$$\alpha_1 = i_1 - 0 \qquad \alpha_2 = i_2 - i_1 \qquad \ldots \qquad \alpha_R = 1 - i_{R-1}$$

We remark that with this representation the nonnegativity (1.2) and the sum-to-one constraints (1.3) still holds, in fact[1]:

$$\alpha_r = i_r - i_{r-1} \geq 0 \quad \text{for } r = 1, \ldots, R$$

$$\sum_{r=1}^{R} \alpha_r = \sum_{r=1}^{R} i_r - i_{r-1} = i_1 - 0 + i_2 - i_1 + \cdots + 1 - i_{R-1} = 1$$

The set of indices represent the *genome* of a possible solution, in the same way the chromosomes completely define an individual in Nature. Each chromosome represents a legal solution to the problem and is composed of a string of genes (i.e. each one of the indeces i_k, with $k = 1, \ldots, R-1$). Once the representation of the model variables has been established, the GA can be applied, according to the scheme presented in 5.

3.3.1 Fitness

The *fitness function*, i.e., the posterior distribution (2.11), provides a measure of performance with respect to a particular set of parameters. The evaluation of a string representing a set of parameters is independent of the evaluation of any other string.

For example, let consider a initial population $\boldsymbol{\alpha}_i^{(0)}$ for $i = 1, \ldots, n$, of $n = 5$ individuals, randomly generated in the interval $[0, 1]$:

$$\begin{array}{lll}
\text{Individual 1} & \boldsymbol{\alpha}_1^{(0)} = & [\ 0.5430 \quad 0.3900 \quad 0.0670\ ]^\top \\
\text{Individual 2} & \boldsymbol{\alpha}_2^{(0)} = & [\ 0.6030 \quad 0.1490 \quad 0.2480\ ]^\top \\
\text{Individual 3} & \boldsymbol{\alpha}_3^{(0)} = & [\ 0.4430 \quad 0.4530 \quad 0.1040\ ]^\top \\
\text{Individual 4} & \boldsymbol{\alpha}_4^{(0)} = & [\ 0.2090 \quad 0.4170 \quad 0.3740\ ]^\top \\
\text{Individual 5} & \boldsymbol{\alpha}_5^{(0)} = & [\ 0.0720 \quad 0.7850 \quad 0.1430\ ]^\top
\end{array}$$

[1] formally we can assign $i_0 = 0$ and $i_R = 1$

Suppose we execute the algorithm in order to evaluate the best fitting individual , i.e., the estimate of the abundance vector, associated with Pixel 1 (see Section 2.4). Given the observation $\boldsymbol{y}$ of Pixel 1 (see Figure 2.3), let $\sigma^{2(0)} = 0.01$ denotes the initial value of the endmember variance. The fitness evaluation according to the posterior (2.11) is:

$$\begin{array}{lll} \text{Fitness of Individual 1} & f(\boldsymbol{\alpha}_1^{(0)}|\boldsymbol{y}, \sigma^{2(0)}) = & 2194.875985 \\ \text{Fitness of Individual 2} & f(\boldsymbol{\alpha}_2^{(0)}|\boldsymbol{y}, \sigma^{2(0)}) = & 2175.101323 \\ \text{Fitness of Individual 3} & f(\boldsymbol{\alpha}_3^{(0)}|\boldsymbol{y}, \sigma^{2(0)}) = & 2196.557458 \\ \text{Fitness of Individual 4} & f(\boldsymbol{\alpha}_4^{(0)}|\boldsymbol{y}, \sigma^{2(0)}) = & 2247.899704 \\ \text{Fitness of Individual 5} & f(\boldsymbol{\alpha}_5^{(0)}|\boldsymbol{y}, \sigma^{2(0)}) = & 1680.394001 \end{array}$$

The fitness function produces a label for each possible solution of the problem, that measure its possibility to survive at the next generation.

The fitness function values constitutes the inputs of the following GAs operators: Copy, Selection, Crossover and Mutation.

The purpose of these operators is to create new solution vectors (*offspring*) by selection, combination or alteration of the current solution vectors that have shown to be good temporary solutions. The new population is further evaluated and tested till termination. If the termination criterion is not met, the population is iteratively operated by the above three operators and evaluated. This procedure is continued until the termination criterion is met.

One cycle of these operations and the subsequent evaluation procedure is known as a *generation* in GAs terminology. The beforementioned operators are described in the following sections.

3.3.2 Selection

If χ is the fraction of the next generation that will be created by crossover, then $1 - \chi$ is the fraction that will be copied intact from this generation to the next. In total then, $(1 - \chi)n$ individuals will be copied over.

The selection of those individual is indiced by the level of fitness achieved. The individuals that show the best fitting property will survive to the next generation. Let suppose that the crossover rate is $\chi = 0.8$ for the example introduced in Section 3.3.1. than $(1-\chi)n = 1$ individual is select as offspring, precisely the one who posess the highest fitness level. For this principle, the Individual 4, characterized by fitness level $f(\boldsymbol{\alpha}_4^{(0)}|\boldsymbol{y}, \sigma^{2(0)}) = 2247.899704$ is chosen as first individual of the next generation, as depicted in Figure 3.4.

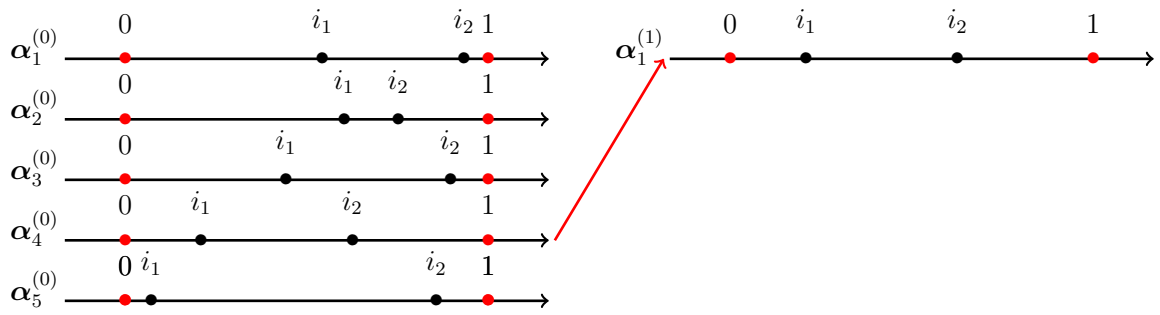


Figure 3.4: Selection of the individual that survive from a genereation to the next ($\chi = 0.8$).

Now we will discuss how the Selection operator chooses the $(\chi - 1)n$ individuals from the current population in order to form a mating pool for reproduction. The basic idea relies on the process of natural selection, which causes those individuals that encode successful structures to produce copies more frequently.

These should be from the fittest individuals of the previous population. There exist a number of reproduction operators in GA literature, but the essential idea in all of them is that the above average strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner.

The Selection operator implemented for the algorithm is the Proportionate Selection operator. This operator selects the individuals for the mating pool with a probability proportional to their fitness. Thus, the individual $\boldsymbol{\alpha}_i$ in the population is selected with a probability proportional to $f(\boldsymbol{\alpha}_i|\boldsymbol{y}, \sigma^2)$. Since the population size is usually kept fixed, the sum of the probability of each string being selected for the mating pools must be one. Therefore, the probability p_i for selecting the individual $\boldsymbol{\alpha}_i$ is:

$$p_i = \frac{f(\boldsymbol{\alpha}_i|\boldsymbol{y}, \sigma^2)}{\sum_{i=1}^{n} f(\boldsymbol{\alpha}_i|\boldsymbol{y}, \sigma^2)} \quad \text{for } i = 1, \ldots, n \tag{3.4}$$

where n is the population size and $\bar{f} = \sum_{i=1}^{n} f(\boldsymbol{\alpha}_i|\boldsymbol{y}, \sigma^2)$ is the average fitness of the population.

One way to implement this selection scheme is to imagine a roulette-wheel with its circumference marked for each individual index i proportionate to the individual's fitness. The roulette-wheel is spun n times, each time selecting an instance of the string chosen by the roulette-wheel pointer.

Since the circumference of the wheel is marked according to the fitness of the individuals, this roulette-wheel mechanism is expected to make $f(\boldsymbol{\alpha}_i|\boldsymbol{y},\sigma^2)/\bar{f}$ copies of the individual $\boldsymbol{\alpha}_i$ in the mating pool.

The Figure 3.5 shows a roulette-wheel for each individuals in the example of Section 3.3.1. Since $\boldsymbol{\alpha}_i$ has higher fitness value than any other, it is expected that the roulette-wheel selection will choose it more than any other individual.

Thereafter, the cumulative probability cp_i of each individual being copied can be calculated by adding the individual probabilities from the top of the list. Formally the cumulative probability cp_i of the individual $\boldsymbol{\alpha}_i$ can be obtained from the probability $p_1, \ldots, p_n$ as follows:

$$cp_i = \sum_{j=1}^{i-1} p_i \qquad \text{for } i = 1, \ldots, n \tag{3.5}$$

Thus, the bottom-most individual in the population should have a cumulative probability values from $cp_i - 1$ to 1. The first individual represents the cumulative values from zero to p_1. Thus, the cumulative probability of any string lies between 0 to 1.

In order to choose n individual for the reproduction, $2n$ random numbers between zero to one are created, since the new offsprings are computed stating from a couple of individuals. Thus, the individual that represents the chosen random number in the cumulative probability range (calculated from the fitness values) is chosen to the mating pool. In this way the string with a higher fitness value will represent a larger range in the cumulative probability values and therefore has a higher probability of being copied. On the other hand, an individual with a smaller fitness value represents a smaller range in cumulative probability values, consequentially has a smaller probability of being copied into the mating pool. In Table 3.2 are computed the cumulative probabilities calculated for the example of Section 3.3.1. Futhermore Figure 3.6 shows how the selection of the individual for the reproduction for this specific problem (blue thin lines), while the first individual has survived from the previous generation (red thick line). We can notice that in reproduction, good individuals are probabilistically assigned in a larger number of copies.

3.3.3 Crossover and mutation

A crossover operator is used to recombine the selected individuals to produce χn better ones. In crossover operation, recombination process creates

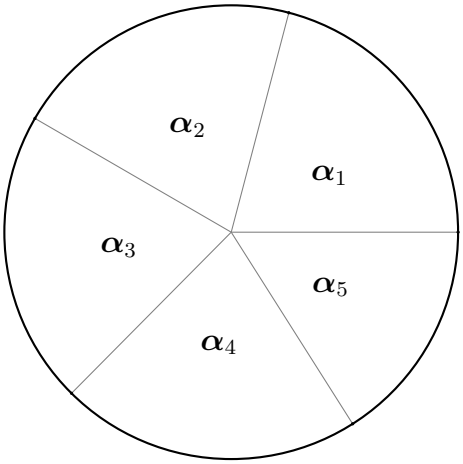


Figure 3.5: A roulette-wheel marked for five individuals according to their fitness values.

individual	cum. probability
$\boldsymbol{\alpha}_1$	0.2091
$\boldsymbol{\alpha}_2$	0.4164
$\boldsymbol{\alpha}_3$	0.6257
$\boldsymbol{\alpha}_4$	0.8399
$\boldsymbol{\alpha}_5$	1

Table 3.2: Cumulative Probability associated with each individuals of the population.

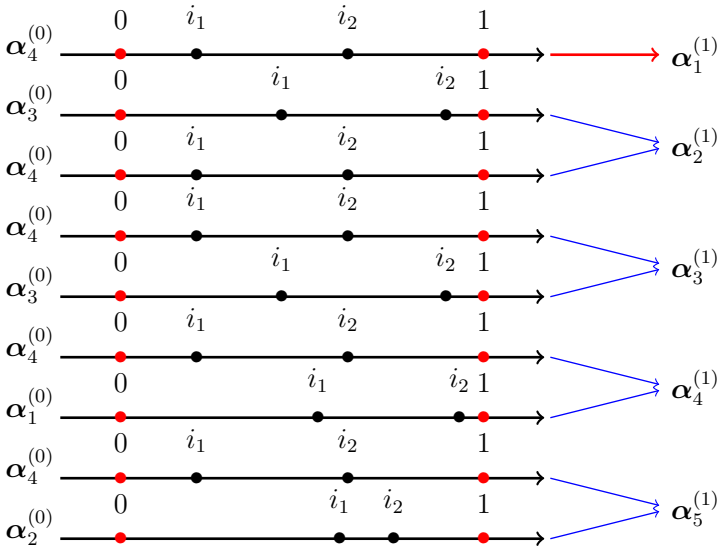


Figure 3.6: Selection of the individual for the reproduction ($\chi = 0.8$).

different individuals in the successive generations by combining material from two individuals of the previous generation. As we see in the previous Section, during the reproduction, good strings in a population are probabilistically assigned a larger number of copies. It is important to note that no new individuals are formed in the Selection phase. In the Crossover, new individuals are created by exchanging information in terms of genes. In particular, in this context we use the single-point crossover. A random position in the coding of the individual is selected, at which swapping occurs.

The two individuals participating at the crossover operation are known as *parent*. It is intuitive from this construction that good genes from the parents can be combined to form a better child genome, if an appropriate site is chosen.

In addition Mutation randomly disturbs this new genetic information. It operates perturbing the position of genes in with small displacements. More precisely, when the indices i_k, $k = 1, \ldots, R-1$ are being copied from the current string to the new string, there is probability that each index may become mutated.

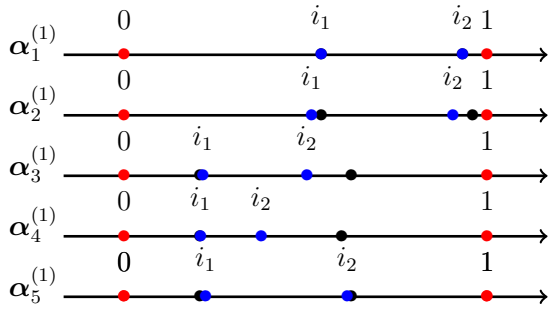


Figure 3.7: Crossover (black dots) and Mutation (blue dots) determine the new individuals of the population.

It is clear from this discussion that the effect of Crossover and Mutation may be detrimental or beneficial. Thus, in order to preserve some of information already present in the former generation, it turns fundamental preserving the best $(1-\chi)n$ individuals (in terms of fitness function) of the former population. Indeed they can not be affected by Crossing and Mutation.

The effect of Crossover and Mutation is exemplified in Figure 3.7: we can see that the new individuals, that inherit the genome from the former population (black dots), are modified my Mutation. As results, the new genes (blue dots) appears displaced with respect of the originals.

Figure 3.8 shows the sampled collected by the algorithm during 500 iterations. We can se a convergence trend of the samples to the real values $\alpha_1 = [0.5 \quad 0.3 \quad 0.2]^\top$.

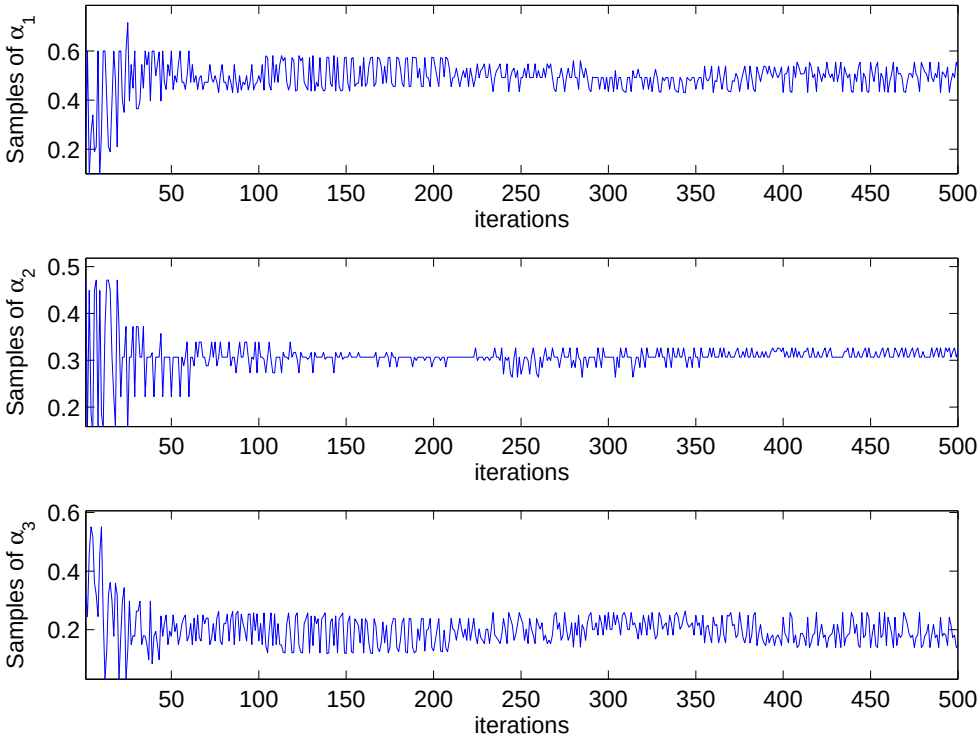


Figure 3.8: parameters samples during 500 iterations. We can se the convergence to the real value $\alpha = [0.5 \quad 0.3 \quad 0.2]^\top$ of Pixel 1.

3.3.4 Discussion

In this Section we want to explain some details about how this algorithm is applied in the hyperspectral unmixing context, more precisely in the sampling parameters procedure.

GAs can be very successful at evolving the fittest individuals. However, a form of stagnation can occur when there is crowding. Crowding occurs when some individual that is fitter than others gets to be selected a lot for the reprodution, so that copies of this individual take over a large part of the population. This reduce the diversity of the population, making hard a progress to be made.

There are numerous ways of reducing crowding. In our implementation we prevent crowding taking a high mutation rate. This is evident in Figure 3.7, where the mutated genome (blue dots) has significantly changed from the original (black dots).
However, a high level of mutation could determine a remarkable variability

in the samples. It could cause the Algorithm not to converge, if we use the convergence criterion exposed in Section 3.1.

For the reasons explained above we decide to adopt the strategy of coupling a high level of mutation with a survival rate $1 - \chi$ that increases during the algorithm execution.

This choice is due to the fact that we can suppose that the fitting property increases as the algorithm iterations, as the parameters space $\mathbb{S}$ is not so wide. With this policy we explore the parameters space at the beginning, and finally we restrict the search in a very close range of the maximum.

3.4 Proxy for the Gaussian distribution

As introduced at the beginning of this Chapter, another way of optimizing an algorithm concearns the linearization problem. In fact, the non-linear computations could slow-down the overall algorithm performance.

For this reason we attempt to give a linearized approximation of the posterior distribution 2.11 which is used both in Algorithm 4 and in Algorithm 5). A linearization is suitable as it does not affect the inner algorithm stucture and leads to estimation errors which are comparable with the algorithm that use 2.11.

From the experimental data 2.4, 2.5 and 2.6 we notice that the distributions that arise from the histograms of all the sets of collected samples resemble *triangular distribuitions*, as depicted in Figure 3.9.

This experimental evidence suggests a *linear* approximation of the Gaussian distribution, for an adequate choice of the parameters.

Triangular distribution seems to be a valid substitute of the Gaussian distribution, despite its simplicity. We remark that, specifically, the triangular distribution has been recently investigated as a proxy for the Beta distribution (see [Johnson 1993]). It confirms its validity for an approximation task.

The triangular distribution constructs a triangular shape from the three input parameters: a minimum (a), most likely (m) and maximum (c) values $a \leq m \leq b$. An example of such distribution is given in Figure 3.10 for different values of the parameters a, b and m.

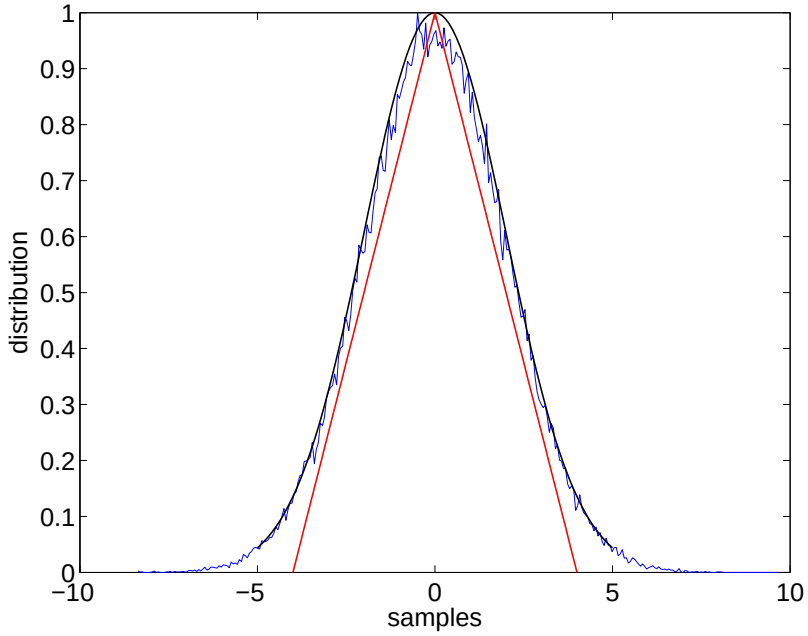


Figure 3.9: histograms of a data set (blue line) approximated by Gaussian distribution (black line) and triangular distribution (red line).

In its most general form, the triangular density function f takes nonzero values in the range $[a, b]$, according to the following expression:

$$f(y) = \begin{cases} \dfrac{2}{b-a}\dfrac{y-a}{m-a} & \text{if } a \leq y \leq m \\ \\ \dfrac{2}{b-a}\dfrac{b-y}{b-m} & \text{if } m \leq y \leq b \end{cases} \tag{3.6}$$

It is again possible to make a simplification of the expression (3.6): since the Gaussian distribution is simmetric, a correct approximation implies the use of a triangular simmetric distribution, also called Simpson's distribution.

If the triangular distribution in (3.6) is simmetric about m, then $m - a = b - m$ and the length of the distribution support is $a + b = 2m$. Then the equation (3.6) reduces to:

$$f_T(y) = \begin{cases} \dfrac{y-a}{(m-a)^2} & \text{if } a \leq y \leq m \\ \\ \dfrac{b-y}{(b-m)^2} & \text{if } m \leq y \leq b \end{cases} \tag{3.7}$$

In the following we refer to the quantity $\sigma_T = \frac{b-a}{2}$ as the *half-base* of the triangular distribution, and to $\mu_T = m$ as its mean value.
With this notation the former equation turns:

$$f_T(y) = \begin{cases} \dfrac{y - \mu_t + \sigma_T}{\sigma_T^2} & \text{if } \mu_T - \sigma \leq y \leq \mu_T \\ \dfrac{\mu_T + \sigma_T - y}{\sigma_T^2} & \text{if } \mu_T \leq y \leq \mu_T + \sigma_T \end{cases} \tag{3.8}$$

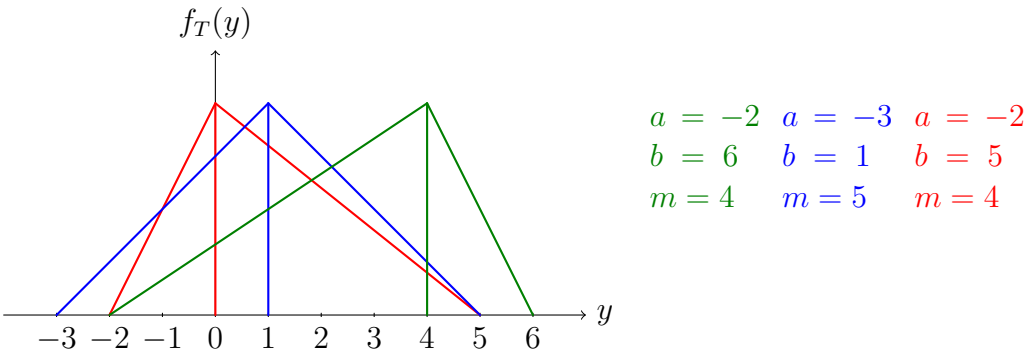


Figure 3.10: different parametrizations of triangular distributions. Symmetric (blue line), right-skewed (red line) and left skewed (green line). the distributions are not normalized.

The triangular distribution is typically used as a subjective description of a phenomenon for which there is only limited sample data, and especially in cases where the relationship between variables is known but data is scarce. It is based on a knowledge of the minimum and maximum and mean value. For these reasons, the triangle distribution has been called a *lack of knowledge* distribution.

These properties fit well to our context, as we need to estimate the (unknown) abundance parameters through sampling. The main issue to handle is related to the adecuate choice of the parameters a, b and m in order to obtain a decent approximation of the posterior distribution in (2.11).

3.4.0.1 The triangular parameters

In this Section we explain how to set the optimum triangular parameters a, b and m in function of the parameter $C(\boldsymbol{\alpha})$ and $\mu(\boldsymbol{\alpha})$) in (2.4) of the Gaussian posterior distribution (2.11).

Let consider first the case of a single-band observation $y = \sum_{r=1}^{R} \boldsymbol{\varepsilon}_r \alpha_r \in \mathbb{R}$, where ε_r are independent Gaussian variable with known means $\boldsymbol{m}_r$ and unknown variance σ^2.

Let $\boldsymbol{M} = [m_1 \dots m_R] \in \mathbb{R}^R$ denote the column vector containing the endmembers means and $\boldsymbol{\alpha} = [\alpha_1 \dots \alpha_R]^\top \in \mathbb{R}^R$ the abundance vector. Then the prior distribution (2.3) of the observation $\boldsymbol{y}$, given the parameters $\boldsymbol{\alpha}, \sigma^2$ reduces to:

$$f(y|\,\boldsymbol{\alpha},\,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2 C(\boldsymbol{\alpha})}} \exp\left(-\frac{(y-\mu(\alpha))^2}{2\sigma^2 C(\boldsymbol{\alpha})}\right) \tag{3.9}$$

where $\mu(\boldsymbol{\alpha}) = \sum_{r=1}^{R} \alpha_r m_r \in \mathbb{R}$ and $C(\boldsymbol{\alpha}) = \sum_{r=1}^{R} \alpha_r^2$.

We suppose that the proxy for the Gaussian distribution (3.9) is a triangular distribution with mean value μ_T equal to $\mu(\boldsymbol{\alpha})$ and half-base σ_T proportional to $C(\boldsymbol{\alpha})$:

$$\begin{cases} \mu_T(\boldsymbol{\alpha}) & = \mu(\boldsymbol{\alpha}) \\ \\ \sigma_T(\boldsymbol{\alpha}) & = k_T \sigma^2 C(\boldsymbol{\alpha}) \end{cases} \tag{3.10}$$

This transformation can be easily extended to the multidimensional case in this way. Let $\boldsymbol{y} \in \mathbb{R}^L$ a pixel observation in L independent bands, as in (2.3). We recall that the prior density function in this case is the *multivariate* Gaussian function:

$$f(\boldsymbol{y}|\,\boldsymbol{\alpha},\,\sigma^2) = \frac{1}{[2\pi\sigma^2 C(\boldsymbol{\alpha})]^{L/2}} \exp\left(-\frac{||\boldsymbol{y}-\boldsymbol{\mu}(\alpha)||^2}{2\sigma^2 C(\boldsymbol{\alpha})}\right) \tag{3.11}$$

In order to obtain the triangular distribution probability associated to a multivariate Gaussian distribution, we first define for each band $i = 1, \dots, L$ the triangular approximation:

$$f_T(y_i|\,\boldsymbol{\alpha},\,\sigma^2) = \begin{cases} \dfrac{y_i - \mu_T(\boldsymbol{\alpha}) + \sigma_T(\boldsymbol{\alpha})}{\sigma_T(\boldsymbol{\alpha})} & \text{if } \mu_T(\boldsymbol{\alpha}) - \sigma_T(\boldsymbol{\alpha}) \le y_i \le \mu_T(\boldsymbol{\alpha}) \\ \\ \dfrac{\mu_T(\boldsymbol{\alpha}) + \sigma_T(\boldsymbol{\alpha}) - y_i}{\sigma_T(\boldsymbol{\alpha})} & \text{if } \mu_T(\boldsymbol{\alpha}) \le y_i \le \mu_T(\boldsymbol{\alpha}) + \sigma_T(\boldsymbol{\alpha}) \end{cases} \tag{3.12}$$

Where the mean value $\mu_T(\boldsymbol{\alpha})$ and the half-base $\sigma_T(\boldsymbol{\alpha})$ of the triangular distribution are obtained as follows:

$$\begin{cases} \mu_T(\boldsymbol{\alpha}) & = \mu(\boldsymbol{\alpha}) \\ \\ \sigma_T(\boldsymbol{\alpha}) & = k_T \sigma^2 \sqrt[L]{C(\boldsymbol{\alpha})} \end{cases} \tag{3.13}$$

From the distribution functions $f_T(y_i|\,\boldsymbol{\alpha},\,\sigma^2)$ we can obtain the overall approximation of the a priori distribution (2.3) as follows:

$$f_T(\boldsymbol{y}|\,\boldsymbol{\alpha},\,\sigma^2) = \sqrt{\sum_{i=1}^{L} f_T(y_i|\,\boldsymbol{\alpha},\,\sigma^2)^2}$$

We can notice that in the case $L = 1$, the trasformations (3.13) and (3.10) are equivalent.

The parameter that characterize the approximation of the Gaussian distribution (2.3) is the ratio between the half-base σ_T and the Gaussian variance $\sigma^2 C(\boldsymbol{\alpha})$:

$$k_T = \frac{\sigma_T(\boldsymbol{\alpha})}{\sigma^2 C(\boldsymbol{\alpha})}$$

The factor k_T is empirically selected in terms of minimization of the Root-Mean-Square Error (RMSE) of the abundance estimate $\hat{\boldsymbol{\alpha}}$ with respecto to the real abundance $\boldsymbol{\alpha}$, defined as follows:

$$\mathrm{RMSE}(\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}) = \sqrt{\frac{1}{R}\sum_{r=1}^{R}(\alpha_r - \hat{\alpha}_r)^2}$$

where α_r represents the real abundance value and $\hat{\alpha}_r$ the estimate, for $r = 1, \ldots, R$.

In Figure 3.11 we can notice that the RMSE is inversely proportional with respect of the factor k_T. From the plot we see that the lowest value of RMSE, is obtained for $k_T = 0.25$. Lower values of k_T may lead to convergence problems. The abundance estimates obtained with triangular proxy of the prior (2.3), given the observation of Pixel 1 (see 2.4) for different values of k_T are shown in Figure 3.12 and 3.13.

As conclusion of this Section, we remark that the linear approximation of the Gaussian distribution with a triangular distribution leads to two advantages. First of all, this one is anallitically easier to handle, as it depends in its general version only on three parametes (only on two, the mean value and the half-base, if it is simmetrical). Then, besides linearization, in the multi-band case (3.12) parallelization is also achieved by the separated processing of each

band, as depicted in Figure 3.14. This constitutes an important improvement in those applications that involve a high number of spectral bands.

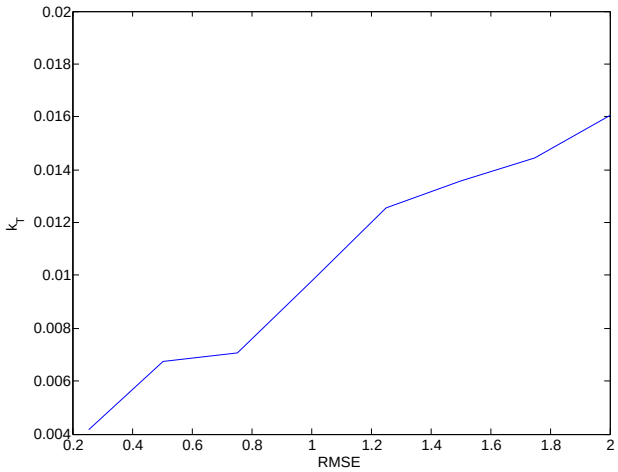


Figure 3.11: histograms of a data set (blue line) approximated by Gaussian distribution (black line) and triangular distribution (red line).

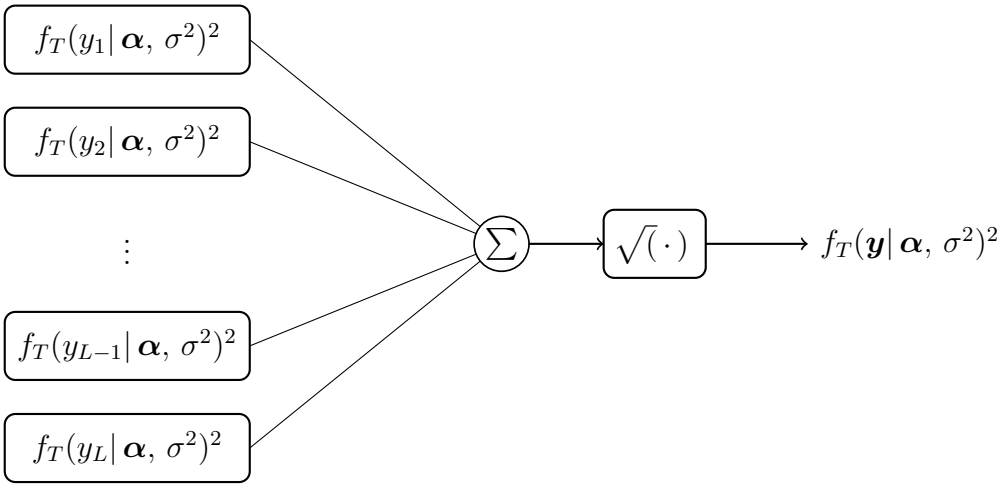


Figure 3.14: Parallelization of the linear unmixing procedure.

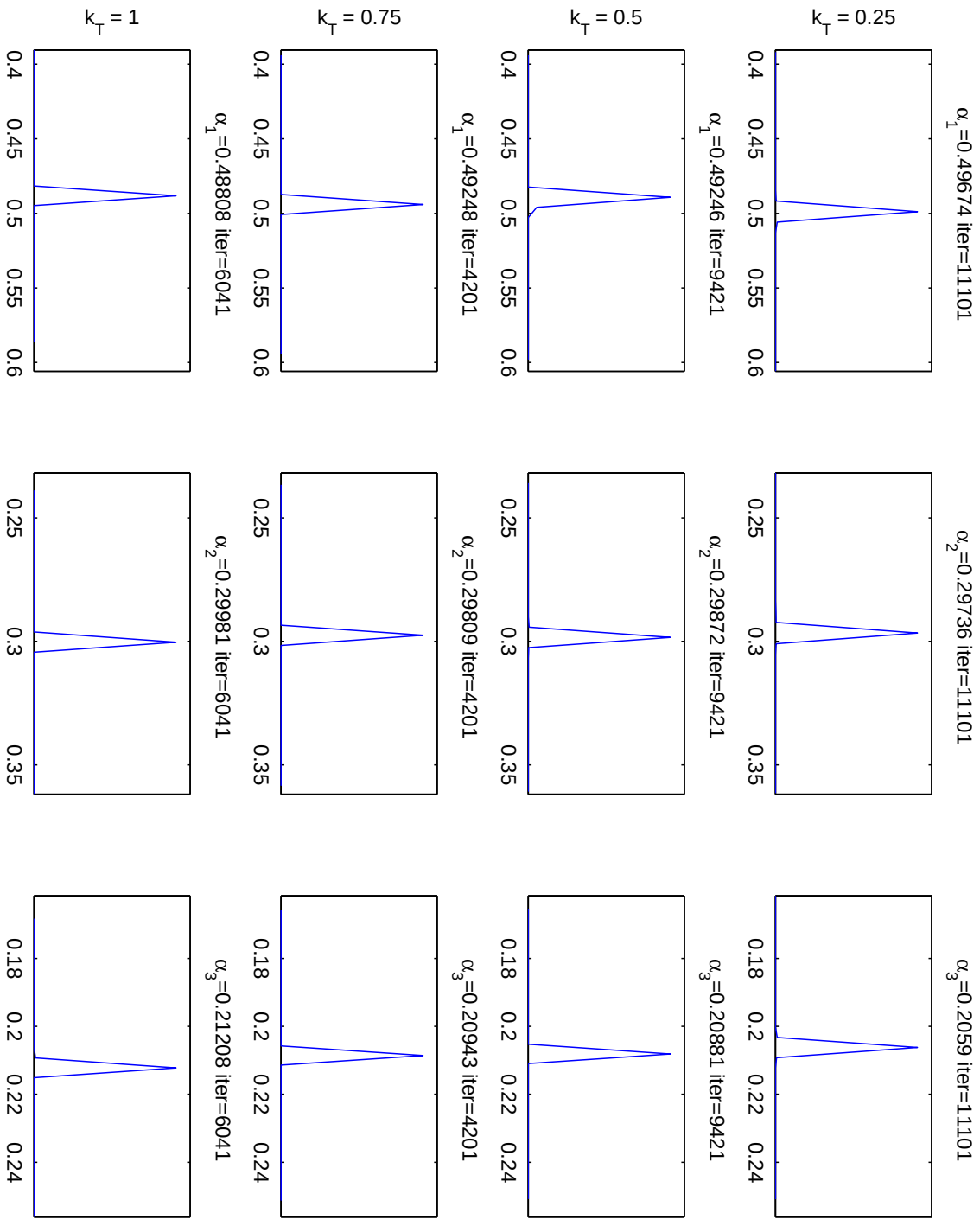


Figure 3.12: Abundance estimation for the Pixel 1 with triangular proxy for differen values of k_T.

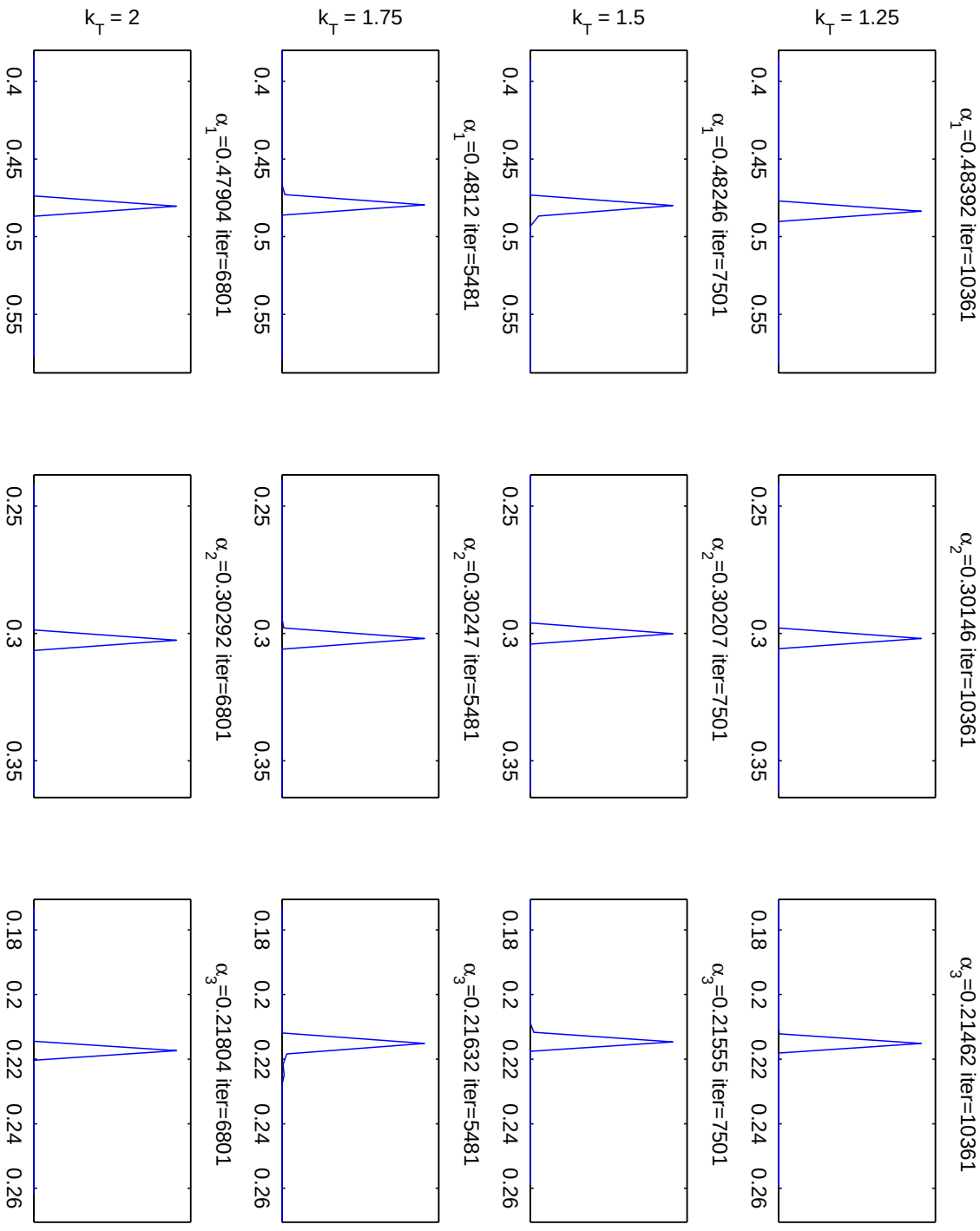


Figure 3.13: Abundance estimation for the Pixel 1 with triangular proxy for differen values of k_T.

3.5 Exploiting spatial correlation in image-domain

In this Section we explain how to exploit spatial correlation in order to process a hyperspectral image. Considering only the spatial dimensions we can represent an image Y of dimension $M \times N$ by a set of pixels $\{\boldsymbol{y}_{ij}\}$ indexed by a pair of coordinates (i, j), $i = 1, \ldots, N$, $j = 1, \ldots, M$, representing its position in the image:

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_{11} & \cdots & \boldsymbol{y}_{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{y}_{M1} & \cdots & \boldsymbol{y}_{MN} \end{bmatrix} \tag{3.14}$$

The Spectral Unmixig procedure provides for each pixel $\boldsymbol{y}_{ij}$, an abundance vector $\boldsymbol{\alpha}_{ij}$, that contains the fraction of the endmembers present in the scene. As for the spectral dimension, we can suppose spatial continuity between contiguous pixels, if the spatial resolution of the sensor is fine enough.

If spatial correlation between contiguous pixels stands, we can suppose spatial correlation between abundances related to contiguous pixel, as the variables $\boldsymbol{y}$ and $\boldsymbol{\alpha}$ are linearly dependent through (2.1).

Inspired by this idea we sequentially process the image pixels estimating their correspondent abundances. Then, the estimates of the current pixel are used as input data for the processing of the following pixel in the sequence. More precisely we will substitute the random generation of the initial abundance vector in Agorithms 3, 4, 5 with the abundance vector calculated in the former step of the sequence.

Indeed, if spatial correlation between pixel stands, the Markov Chain Monte Carlo will converge rapidly to the stationary distribution as it starts from favorable initial condititons.

The image processing method that we adopt in this work consists in exploring the image pixels column by column from the top to the bottom of the image. This choice is arbitrary, as we can scan the image horizontally row by row, from the left to the right, as depicted in Figure 3.15. The favorable effects about using one or another technique is related with the specifical topology of the scene.

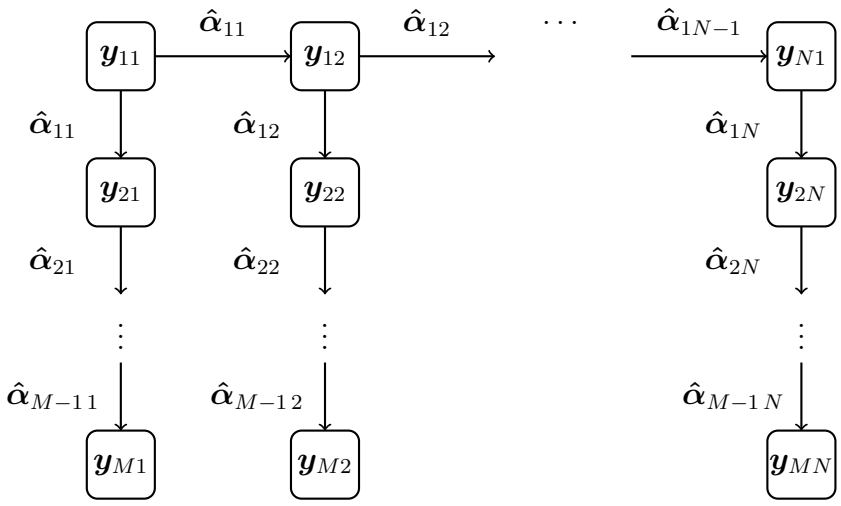


Figure 3.15: Sequential processing of the image exploiting the spatial correlation.

Chapter 4

Validation an discussion

Contents

In this Chapter we conduct a quantitative and comparative experimental validation of the main contributions of this work introduced in previous Chapters: introduction of an innovative sampling technique using GAs, linear proxy of the gaussian prior distribution (2.3) and explotation of spatial correlation for the hyperspectral image unmixing.

For the sake of simplicity we introduce a concise notation to distiguish the different characteristics of the tested algorithms. The reference Algorithm A_0 is [Eches 2010], presented in Chapter 2. The Algorithms A_1, A_2, A_3, A_4 are characterized by the features presented in Table 4.1. In all of them the convergence condition explained in 3.1 is applied. Further, we explore the effects of introducing the sample step based on GAs (A_2), and the proxy of the Gaussian function with the triangle distribution (A_3), and the combined effects of the two modifications in (A_4).

The Chapter is structured as follows. First, we introduce the quantitative metrics that will be used to carry the experimental validation. Then, we present different data sets used for the experiments. These data comprise both synthetic and real hyperspectral data sets, where the procedure for generating the synthetic data is described in detail in this Chapter. Then, we present the results obtained by the approaches presented in this work with the considered data. The Chapter concludes with a discussion on the main observations

	A_0	A_1	A_2	A_3	A_4
Convergence Criterion		✓	✓	✓	✓
GA			✓		✓
Tiangular Proxy				✓	✓

Table 4.1: Algorithms features comparison chart.

resulting from the conducted experiments and also the aspects introduced by these techniques.

4.1 Quantitative metrics

In this section we describe the different metrics and measurements used in order to validate the results provided by the new techniques presented in this work. In order to validate the accuracy of the spectral unmixing process we use two metrics. Their definition are described in the following subsections.

4.1.0.2 Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) intends to evaluate the reconstruction quality, comparing the real pixel value and its estimation by reconstruction. Since ground-truth abundances are very difficult to be obtained in practice, this metric provides a way to use the original hyperspectral image as a reference for the evaluation.

Low values of RMSE mean that the pixel is well described as a linear combination of the endmembers and its abundances. However, a high RMSE means that the linear combination of the endmembers and the abundances does not properly reconstruct the original hyperspectral scene.

This may be caused by a wrong choice of the endmembers, which do not describe the spectral singularity of this pixel as a linear combination of these endmembers, or to an inappropriate estimation of endmembers abundances.

In the case of real data sets, errors may also arise, due to the fact that the linear mixture model may not properly characterize nonlinear mixing effects that often happen in Nature.

Let $\boldsymbol{Y}$ denote the original image and $\hat{\boldsymbol{Y}}$ the correspondent estimate. Suppose that $\hat{\boldsymbol{Y}}$ consists of $M \times N$ pixels observed in L spectral bands. The RMSE of the estimate $\hat{\boldsymbol{Y}}$ with respect to $\boldsymbol{Y}$ is:

$$\mathrm{RMSE}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sqrt{\frac{1}{L} \mid\mid \boldsymbol{y}_{ij} - \hat{\boldsymbol{y}}_{ij} \mid\mid^2} \tag{4.1}$$

4.1.0.3 Signal-to-Reconstruction Error (SRE)

The Signal-to-Reconstruction Error (SRE) measures the quality of the reconstruction of a spectral mixture in terms of abundance estimates. This metric gives more information regarding the power of the error in relation with the power of the abundance in comparison to the RMSE.

High values of SRE means that the algorithm provides good abundance estimate of a pixel. On the contrary low values implies bad estimates.

The evaluation of this parameter only involves the synthetic images, as ground-truth abundance map are not available for real data. Let $\boldsymbol{\alpha}_{ij}$ denote the abundance vector of the pixel $\boldsymbol{y}_{ij}$, and suppose the image consists of $N \times M$ pixel. Then the SRE can be obtained as follows:

$$\mathrm{SRE} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \mid\mid \boldsymbol{\alpha}_{ij} \mid\mid^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} \mid\mid \boldsymbol{\alpha}_{ij} - \hat{\boldsymbol{\alpha}}_{ij} \mid\mid^2} \tag{4.2}$$

It is measured in dB: $\mathrm{SRE(dB)} = 10 \log_{10} \mathrm{SRE}$.

4.2 Hyperspectral data used in the experiments

In our experiments we have used both synthetic and real hyperspectral datasets. The primary reason for the use of synthetic imagery as a complement to real data analysis is that all details of the simulated images are known. These details can be efficiently investigated because they can be manipulated individually and precisely.

As a result, algorithms performance can be examined in a controlled manner. In the following we describe the synthetic and real hyperspectral data sets used in our experiments.

4.2.1 Synthetic data

A database of sixteen 30×30 pixels synthetic hyperspectral images has been created. The endmemembers have been selected from the *U.S. Geological Survey Library* [USGS 2012]. the spectral signatures occupy $L = 224$ bands and a range of wavelength from $0.2\mu m$ to $2.6\mu m$. In particular, to form the

synthetic images library we selected $R = 3$ spectral signatures of different minerals, namely Carlannite, Ammonioalunite and Biotite (see Figure 4.1).

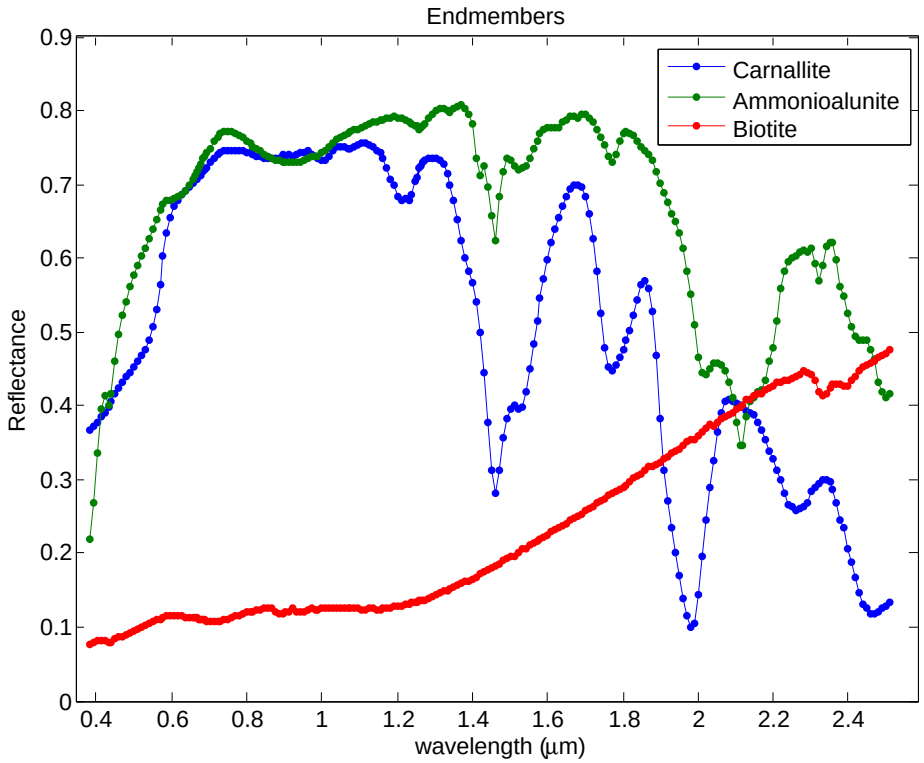


Figure 4.1: Spectral reflectance of Carnallite, Ammonioalunite, and Biotite obtained from [USGS 2012].

The abundances maps are created using a *cellular automaton*. This system makes possible controlling the level of mixture and location of the materials in a fully constrained fashion. Even, the abundance maps obtained in this way seems to have fractal properties.

We are interested in obtaining fractal features in synthetic data because several natural objects can be approximated by fractals to a certain degree, including clouds, mountain ranges, coastlines, vegetables, etc. thus providing a baseline for simulating spatial patterns often found in Nature.

The use of cellular automaton consists of an iterative process where each pixel in each iteration has 3 possible actions with 3 different probabilities. These three actions are: **mixing**, **exchange** and **no-operation**.

Mixing consists of mixing the abundance vector of the given pixel

with some random neighbour pixels. Exchange consists of changing the value of the abundance vector by the value of one of its neighbours. No-operation consists of keeping the previous abundance values of the given pixel.

We can define the endmembers distribution seed, the different probabilities for each action, the size of the neighborhood window and the total number of iterations. With the probabilities it is possible to control the level of roughness and even the level of mixture in a qualitative way. For instance, if the probability of mixing is high the resulted abundances map will be very mixed, and if the probability of exchanging is high the resulted abundances map will have high spatial roughness. Furthermore, the level of mixture can be controlled running the automaton for a fixed number of iteration. A high number of iterations correspods to a homogeneous scene, in which the level of mixture is high. On the contrary a low iteration number implies a high level of "roughness" in the scene.

The used probabilities for creating the synthetic images were: 0.2 for mixing, 0.5 for exchange and 0.3 for no-operation. Four type of images have been created by running the automaton for 50, 100, 150 and 200 iterations.

Then, zero-mean Gaussian noise was added to the four synthetic scenes in different signal to noise ratios (SNRs) 10, 20, 40 and 60 dB to simulate noisen contributions from ambient and instrumental sources.

The different synthetic images will be connoted by the level of noise (SNR = 10, 20, 40, 60 dB) and the number of iteration of the cellular automaton (CONT = 50, 100, 150, 200).

In Figure 4.2 are shown the abundance maps obtained by the cellular atomaton with $R = 3$ endmembers. We notice that for a low iterations level (first row) the scene appears "spotted" because the distribution seed have been scarcely mixed. Instead, increasing the iteration number, the scene appears quite homogeneous.

Then the abundance maps are multiplied for the correspondent endmembers spectra in order to obtain the synthetic image. In Figure 4.3 we can appreciate the effect of adding the Gaussian noise to various scenes: from the left to the right the noise level decreases, and the scene appears less corrupted. In Figure 4.4 we report the plot of the pixel spectra (regardless the spatial coordinates) appreciating the corruption caused by the noise.

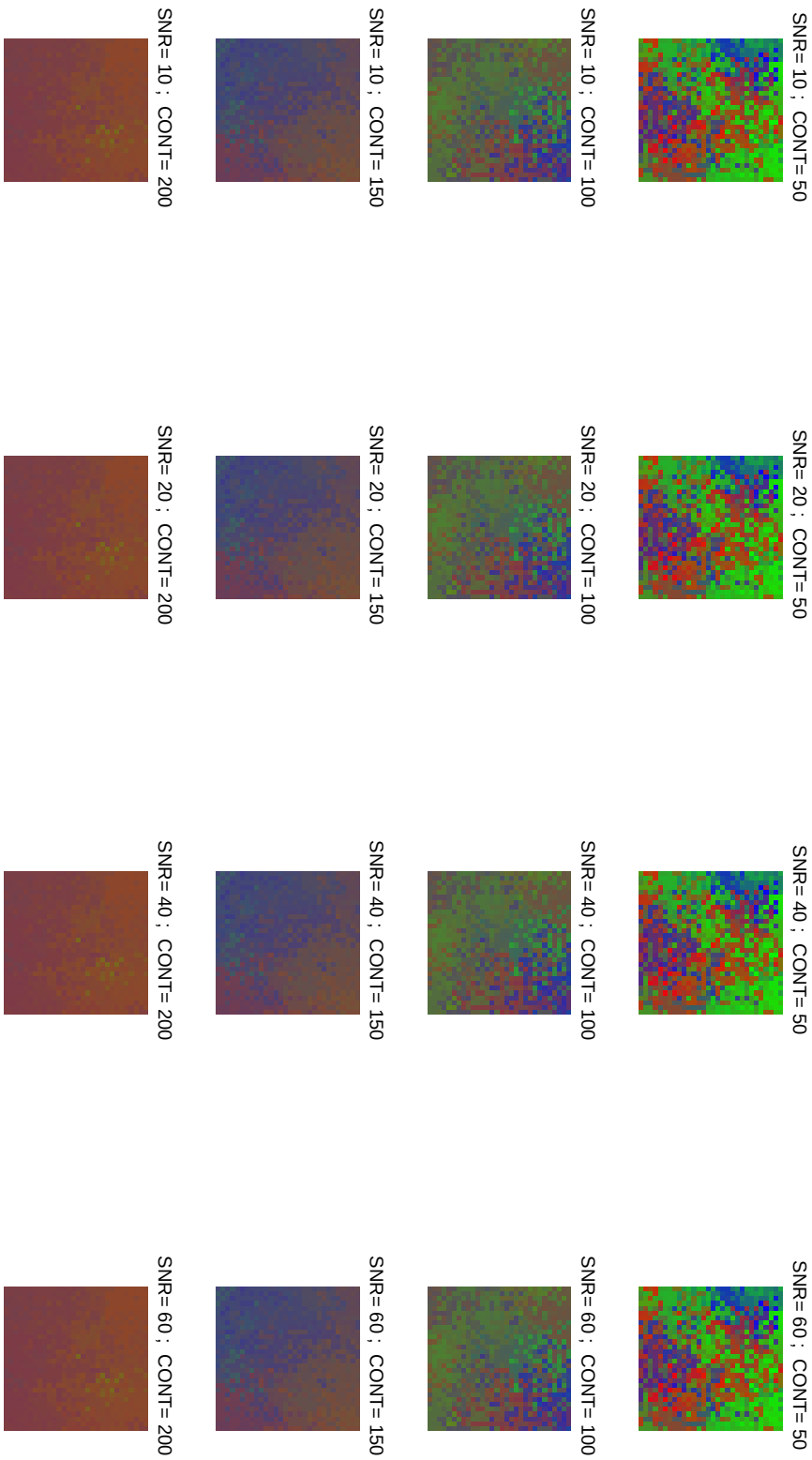


Figure 4.2: Abundances maps of the synthetic data library.

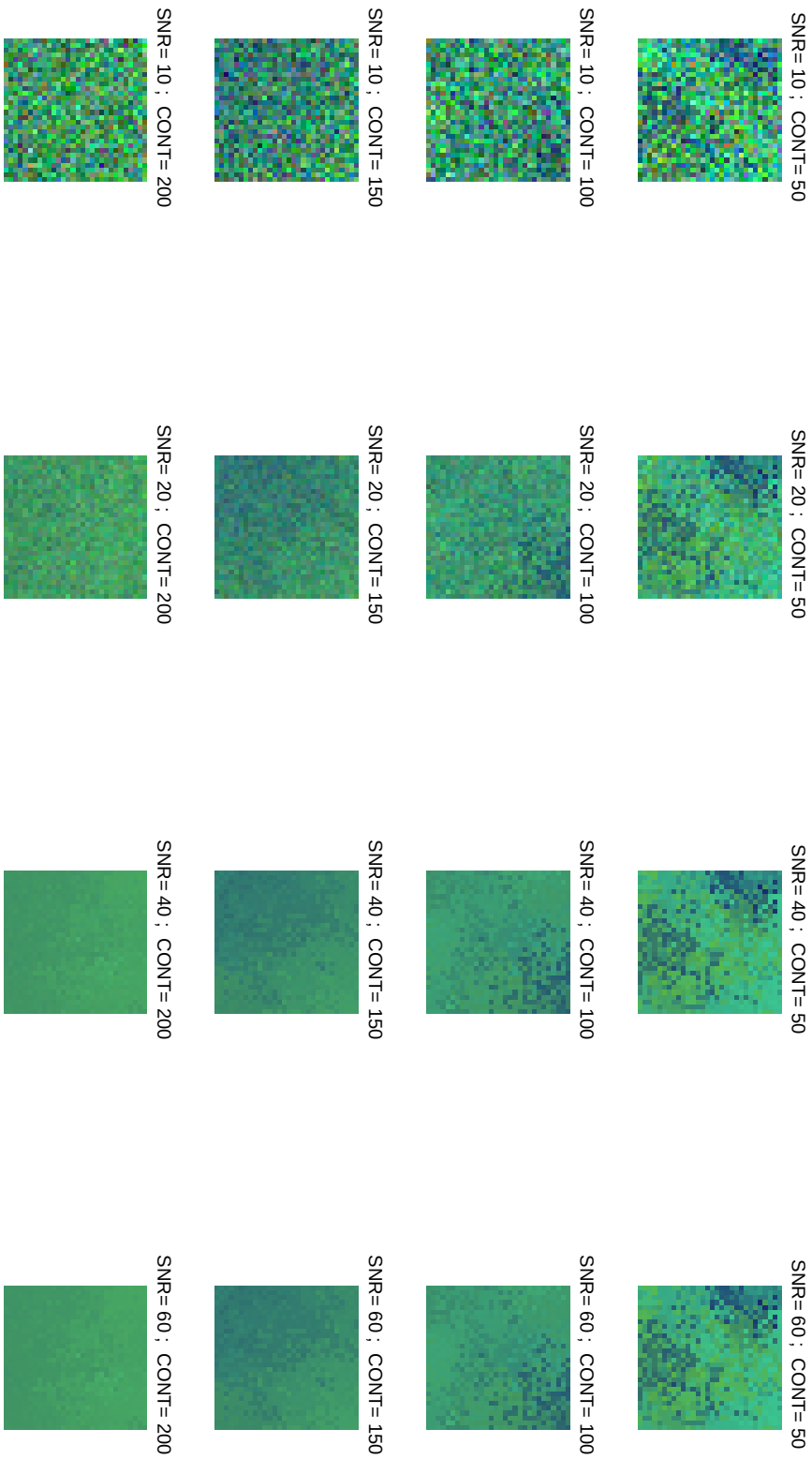


Figure 4.3: Three-channel plot of the synthetic data library.

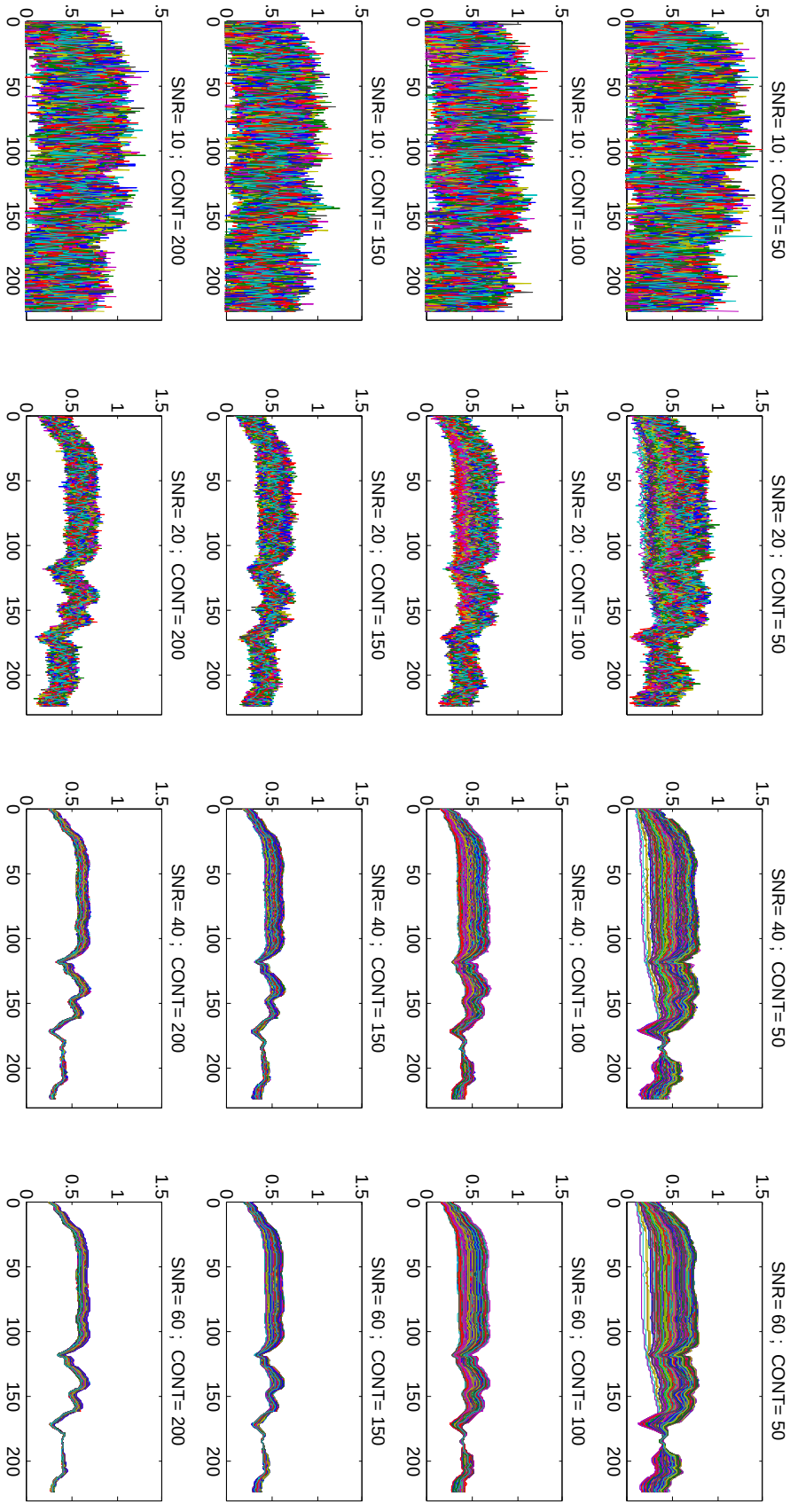


Figure 4.4: pixels spectra of the synthetic data library.

4.2.2 Real hyperspectral data

In addition to synthetic data experiments, it is important to justify algorithms performance using real data sets. However, the evaluation of spectral unmixing techniques is difficult because there are few hyperspectral scenes with reliable reference information.

This is due to the fact that, in practice, it is difficult to collect ground-truth information of abundances at a sub-pixel level. However, the well-known AVIRIS Cuprite scene has been widely used to evaluate spectral unmixing algorithms. The scene, available online in reflectance after atmospheric correction, is characterized by the availability of some very reliable reference information available from USGS.

Specifically, the portion used in experiments corresponds to a 50×50 pixel subset of the image 4.6.

The scene comprises $L = 224$ spectral bands between $0.4\mu m$ and $2.5\mu m$, with full width at half maximum of $10nm$ and spatial resolution of 20 meters per pixel. Prior to the analysis, several bands were removed due to water absorption and low SNR in those bands, leaving a total of 188 reflectance channels to be used in the experiments. The Cuprite site is well understood mineralogically, and has several exposed minerals of interest, all included in the USGS library considered in experiments, denoted *splib065* and released in September 2007. In our experiments, we use spectra obtained from this library to validate endmembers extraction algorithms. For illustrative purposes, Figure 4.5 shows a mineral map produced in 1995 by USGS.

4.3 Evaluation of spectral unmixing algorithms

In this Section we present an evaluation of the spectral unmixing techniques developed in this work. First, we illustrate the results obtained with synthetic data and then we conduct an evaluation using real data. The metrics used for evaluation purposes are the RMSE (4.1) and the SRE (4.2).

4.3.1 Results with synthetic data

This section describes the results obtained for Spectral Unmixing algorithms using the synthetic data set generating using tha cellular automaton.

Before describing our experiments, it is first important to address the parameter values used for our proposed algorithms:

- A_0: fixed iterations number Nmc=25000 and burn-iterations Nbi=5000.

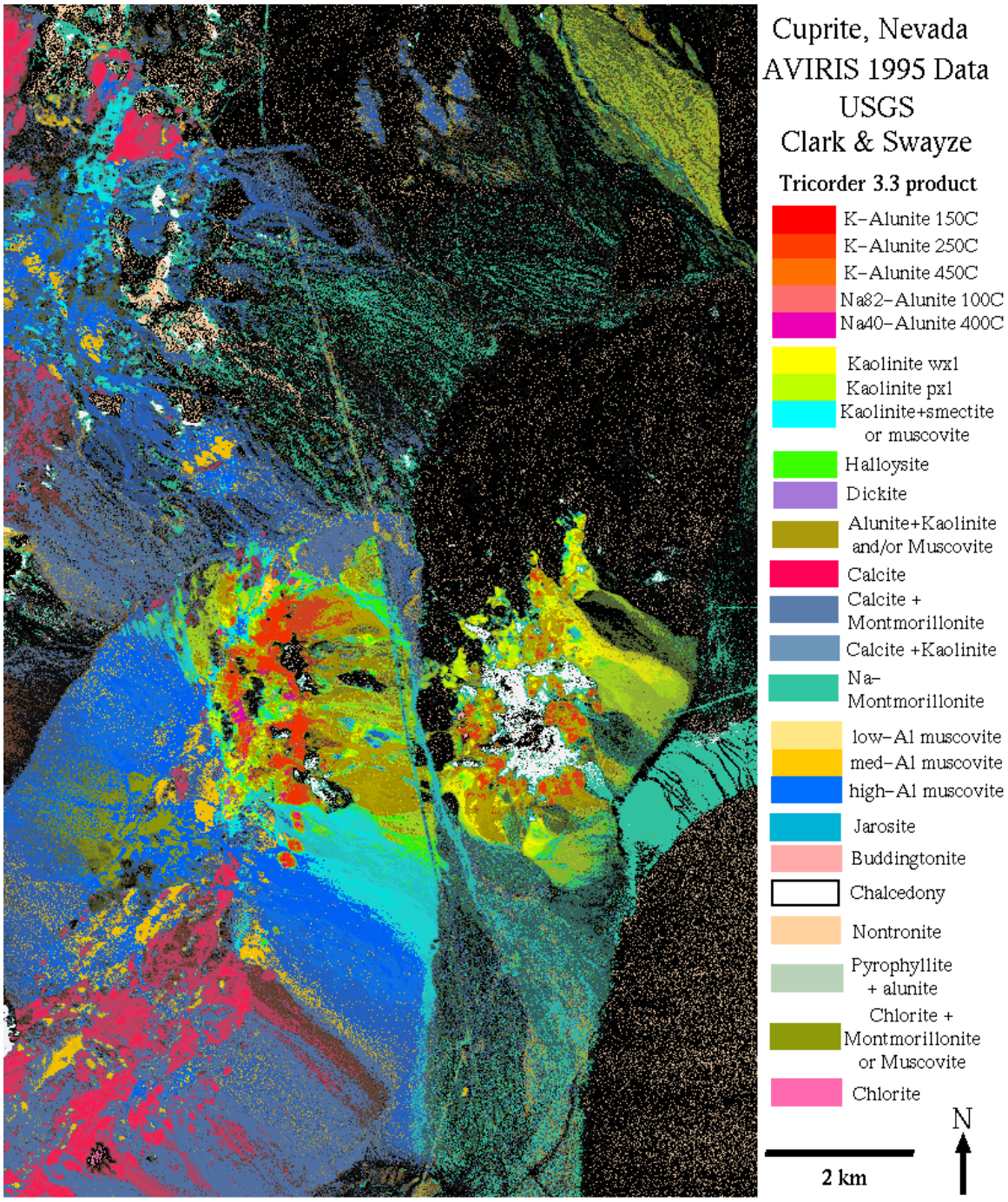


Figure 4.5: USGS map showing the location of different minerals in the Cuprite mining district in Nevada.

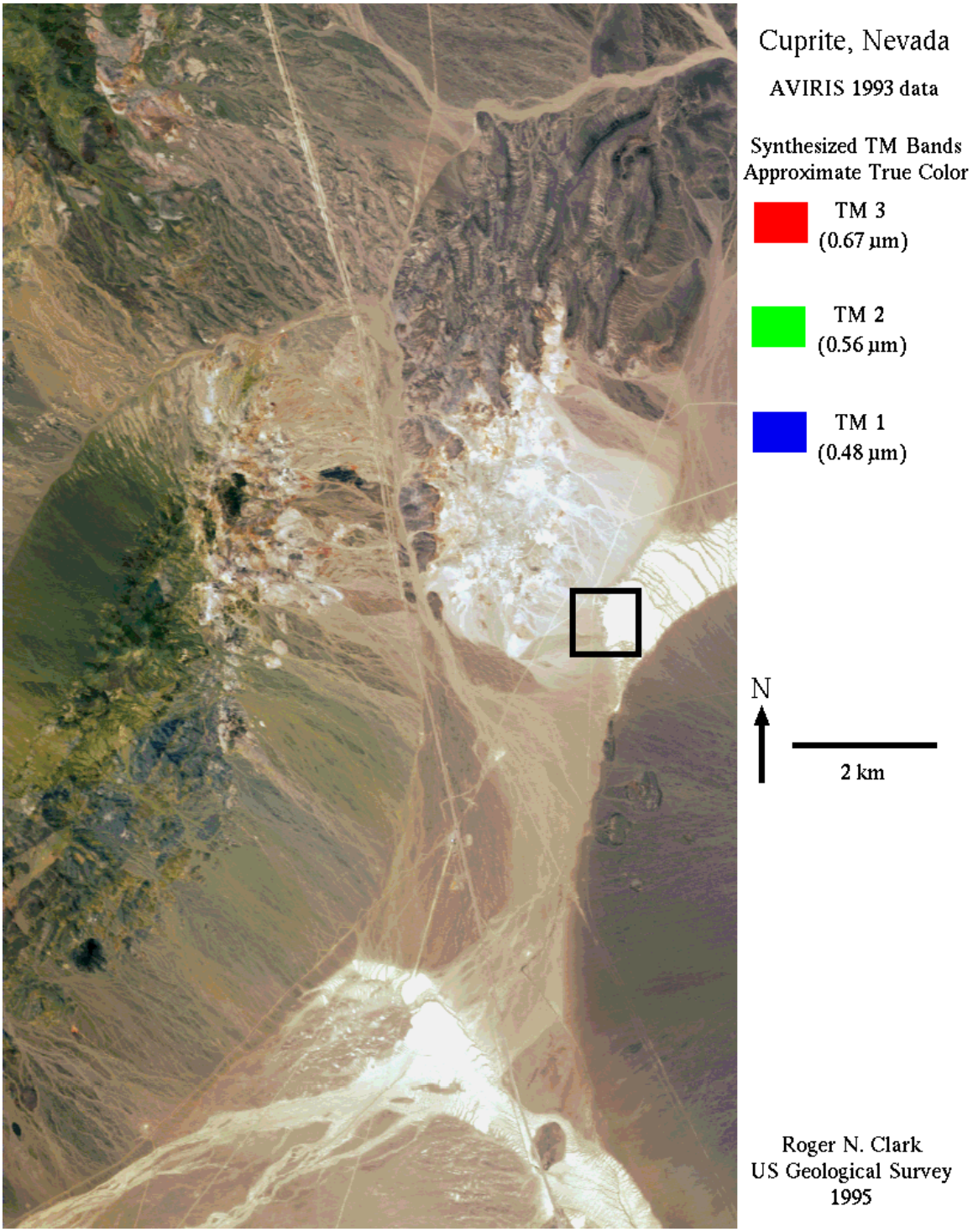


Figure 4.6: USGS map showing the Cuprite mining district, Nevada. The black square depicts the area used in the experiments.

- A_1: convergence threshold $\varepsilon = 10^{-7}$.
- A_2: convergence threshold $\varepsilon = 10^{-7}$, number of individuals $n = 10$, crossover rate initial value $\chi_0 = 0.8$, which decrease of 0.1 every 100 iterations.
- A_3: convergence threshold $\varepsilon = 10^{-7}$, approximation factor $k_T = 0.25$.
- A_4: convergence threshold $\varepsilon = 10^{-7}$, approximation factor $k_T = 0.25$, number of individuals $n = 10$, crossover rate initial value $\chi_0 = 0.8$, which decrease of 0.1 every 100 iterations.

In particular, for the Algorithms A_2 and A_4 we provide two different versions, namely A_2v1 and A_2v2. The random population generated in the initialization of the Algorithm 5 (see Chapter 3) is contained in an interval of length $\delta = 10^{-4}$ and $\delta = 10^{-7}$ respectively centered in the previous result of the unmixing processing, according to the sequential order in (3.15).

The simulations results are presented in Figures 4.7, 4.8,4.9, 4.10, 4.11, 4.12 and 4.13.

In Figure 4.10 and 4.11 we can compare the algorithms performance in terms of RMSE through a scatterplot, that correlates the iteration numbers with the RMSE values. We can notice that, for a high level of noise (i.e, SNR = 10 dB) the algorithm A_2v1 leads to RMSE values that are comparable with A_0 but with a lower iterations number. This indicates robustness of the approches used in that algorithm in high noise condtion.

As the noise level decrease, i.e., for SNR= 40, 60 dB the algorithm A_4 prove its effectiveness, obtaining RMSE values comparable with the reference A_0. We notice that the algorithm A_1 provides higher RMSE obtained for higher iterations number with respect to A_4, proving that the linearization does not affect the algorithm estimation capability. On the contrary, it provides better estimates with lower computational cost.

The synergy of linearization and GA is not effective in this context. From the simulation results, we can notice that the two versions of A_4 do not provide results comparable with other algorithms.

The SRE values are in full agreement with the discussion explained before. In particular in Figure 4.13 we can appreciate how the algorithm A_4 provides even better results in terms of reconstruction error than the others.

Although the results obtained with synthetic hyperspectral scenes are very encouraging, further experiments with a real hyperspectral scene are conducted in the following subsection in order to fully substantiate the advantages that can be obtained by using the Genetic Algorithm and the linearization procedure.

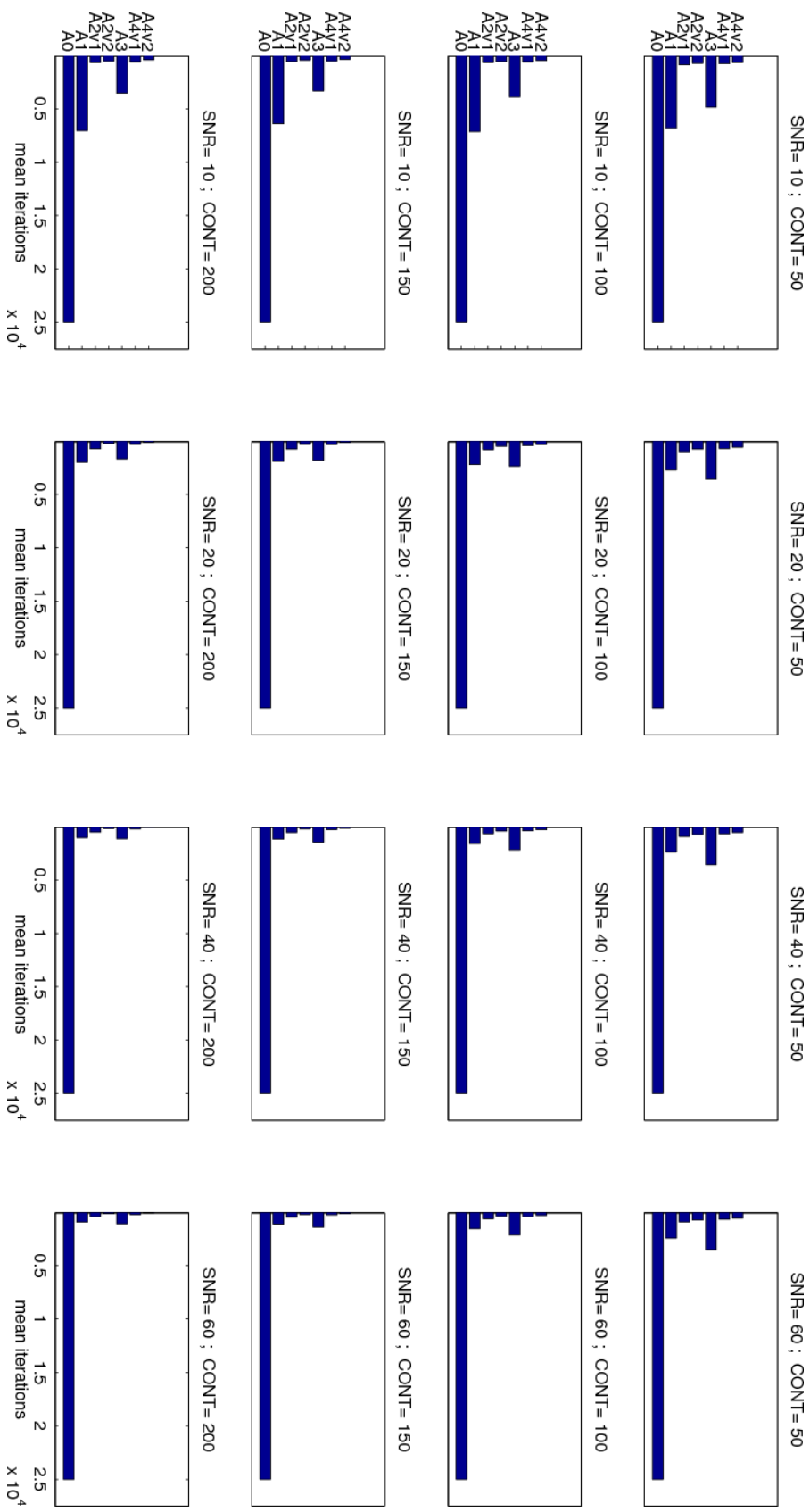


Figure 4.7: Bar diagram of the iterations number obtained from experiments with synthetic data. In each rectangle, related to a synthetic figure in the spectral library, the performance of the Algorithms in terms of iteration is compared.

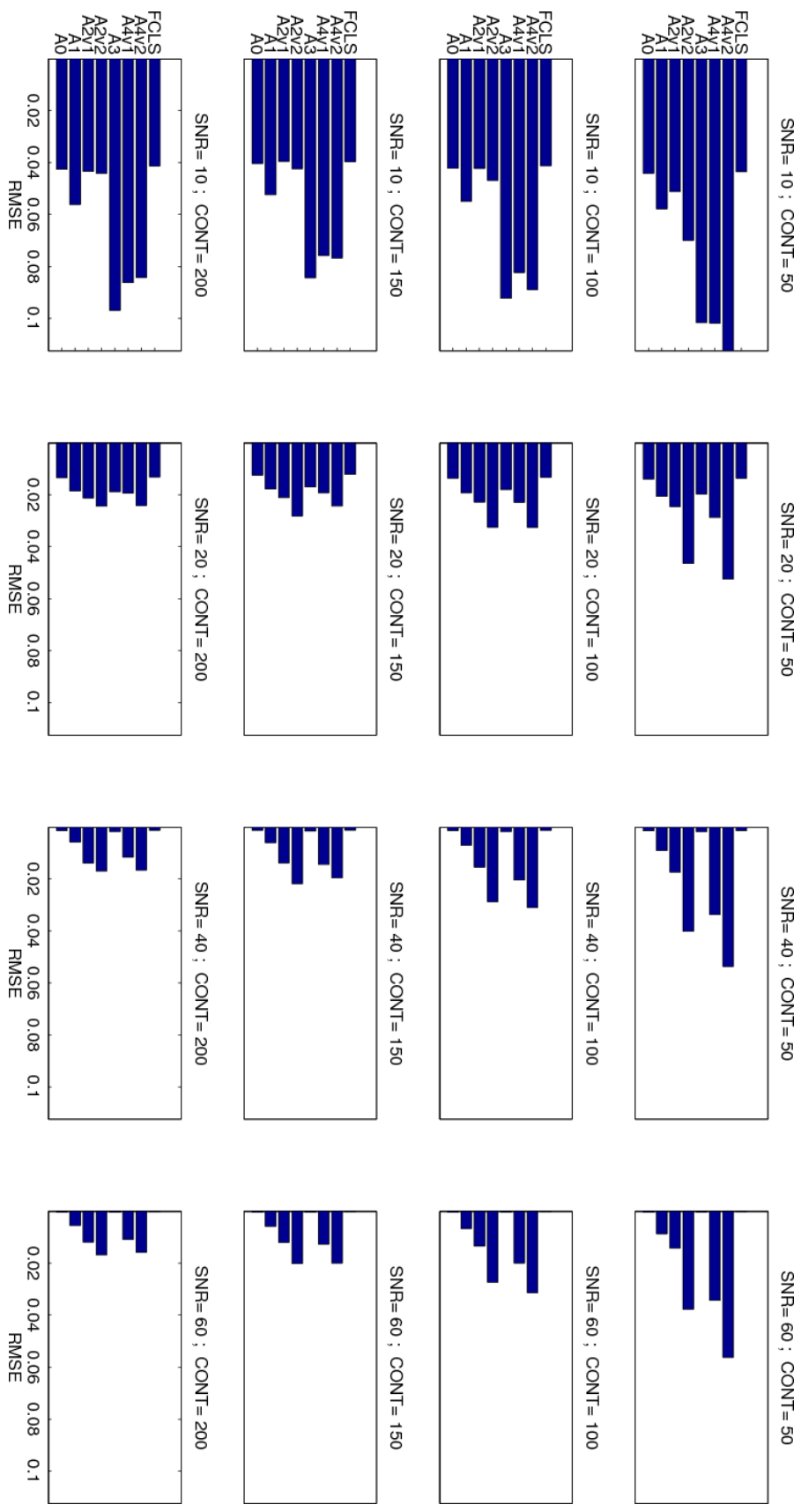


Figure 4.8: Bar diagram of the RMSE values obtained from experiments with synthetic data. In each rectangle, related to a synthetic figure in the spectral library, the performance of the Algorithms in terms of RMSE is compared.

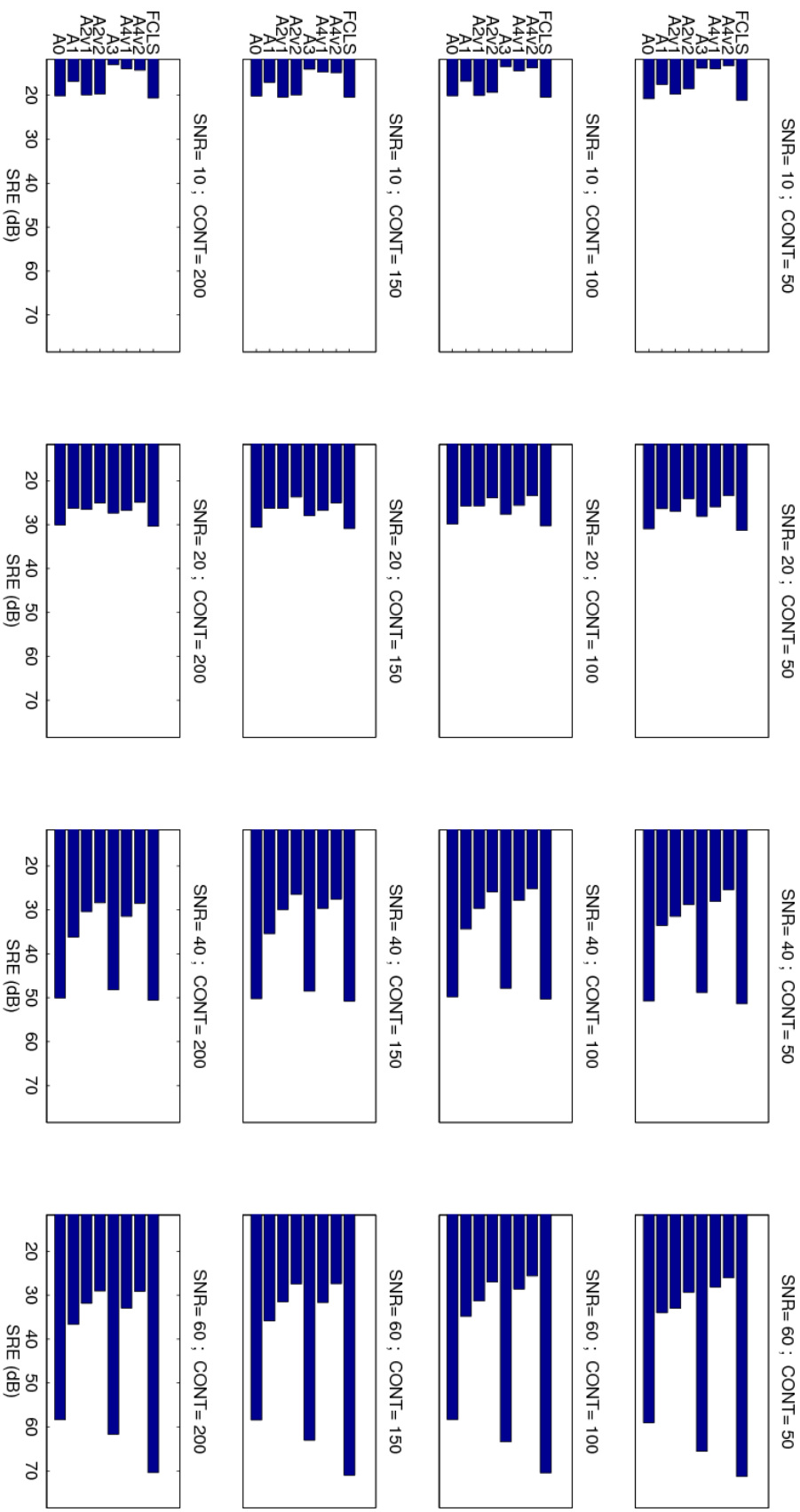


Figure 4.9: Bar diagram of the SRE values obtained from experiments with synthetic data. In each rectangle, related to a synthetic figure in the spectral library, the performance of the Algorithms in terms of SRE is compared.

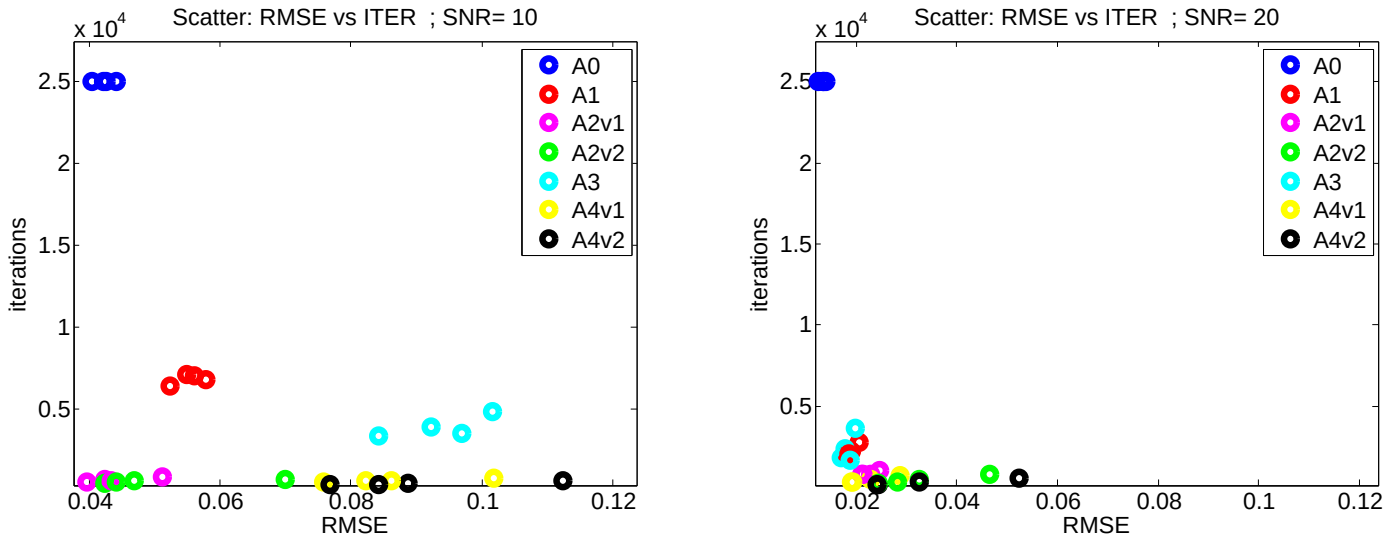


Figure 4.10: Scatter Plot of iterations and RMSE for SNR = 10dB (left) and for SNR = 20dB (right). The images grouped by colour are characterized for the same CONT number.

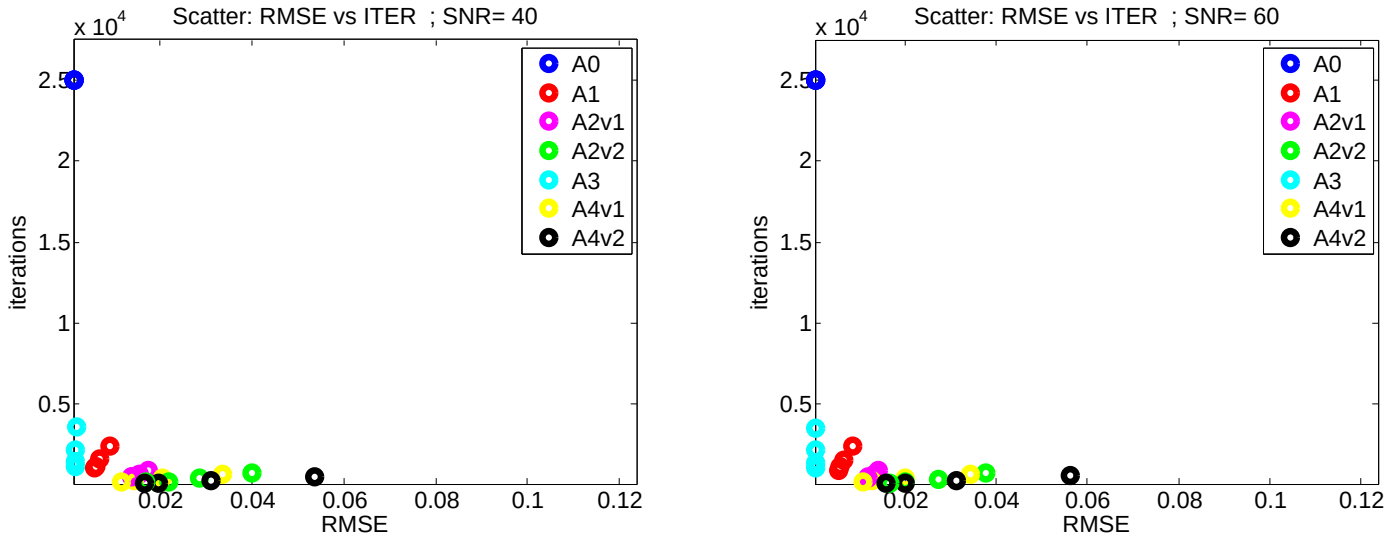


Figure 4.11: Scatter Plot of iterations and RMSE for SNR = 40dB (left) and for SNR = 60dB (right). The images grouped by colour are characterized for the same CONT number.

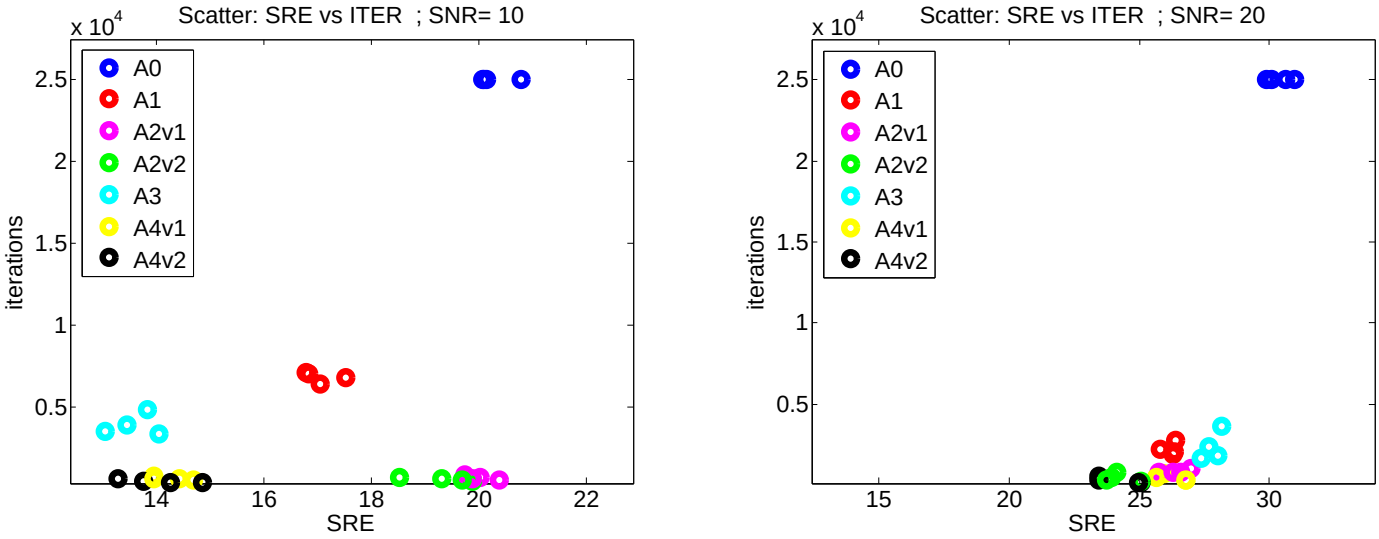


Figure 4.12: Scatter Plot of iterations and SRE for SNR = 10dB (left) and for SNR = 20dB (right). The images grouped by colour are characterized for the same CONT number.

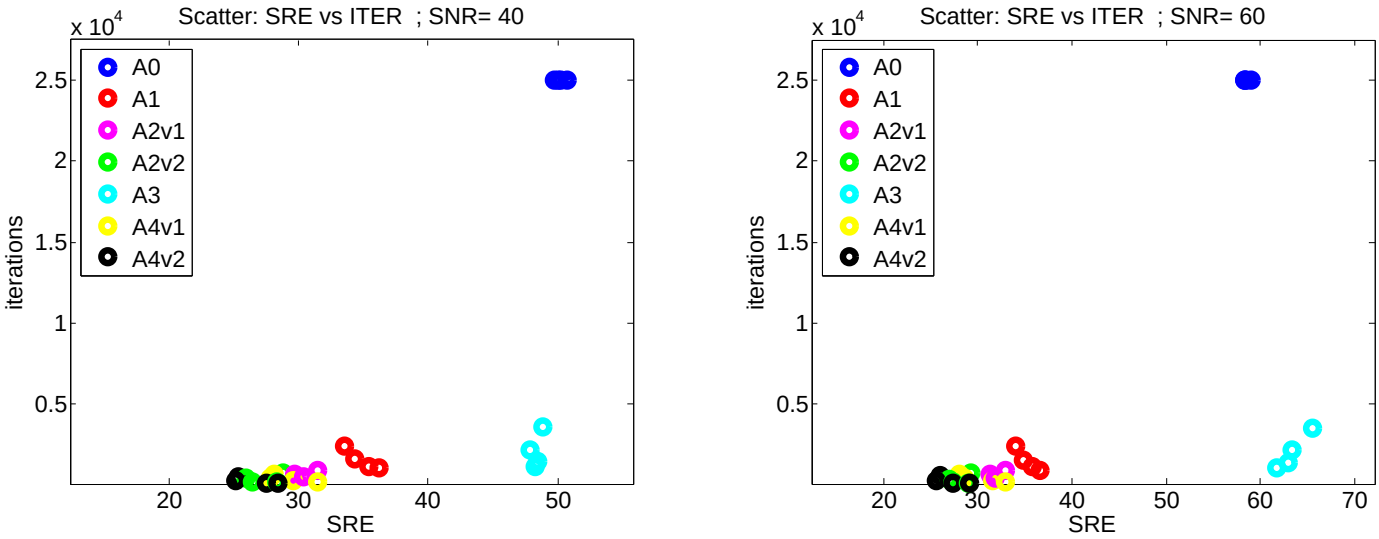


Figure 4.13: Scatter Plot of iterations and SRE for SNR = 40dB (left) and for SNR = 60dB (right). The images grouped by colour are characterized for the same CONT number.

4.3.2 Results with real data

In this section we will describe the experiments performed with the real dataset collected by AVARIS over the Cuprite mining district, in particular for the 50×50 pixels selected area. For this type of experiments, the abundances estimation algorithms $A_1 - A_4$ are coupled with the VCA endmembers extraction algorithm, as they required the *a priori* knowledge of the deterministic edmembers $\boldsymbol{m}_1, \ldots, \boldsymbol{m}_R$ contained in the matrix $\boldsymbol{M}$. Consequently, the actual endmember matrix $\boldsymbol{M}$ is also used for computing the RMSE end SRE values associated to each algorithm. The $R = 3$ endmembers extracted by the VCA algorithm are shown in Figure 4.14, and have been used as mean vectors $\boldsymbol{m}_1$, $\boldsymbol{m}_2$ and $\boldsymbol{m}_3$. We can notice the presence of gaps in the spectral signature. They correspond to wavelength ranges at which the atmosphere absorbs so much light than no reliable signal is received from the surface. The experiments are conducted by setting the same parameters simulation values as for the synthetic data set.

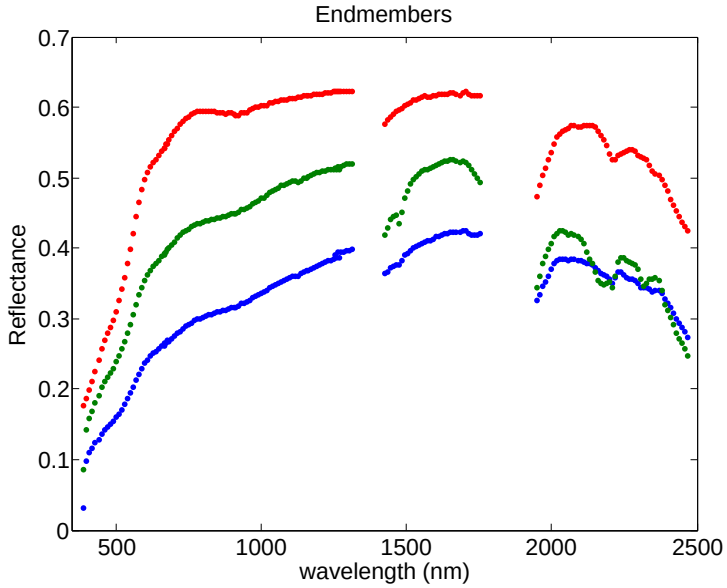


Figure 4.14: Spectral reflectance of the endmembers extracted by VCA algorithm.

The experiments results are displayed through the abundances colormap depicted in Figure 4.15. Each abundance component α_1, α_2, α_3 is characterized by a RGB color, as the correspondent endmember in Figure 4.14. We can appreciate the presence of three homogeneous areas, each one characterized by the predominance of one endmember. The algorithms performance are illustrated in Figure 4.17. The algorithm A_1 is not present in the results as

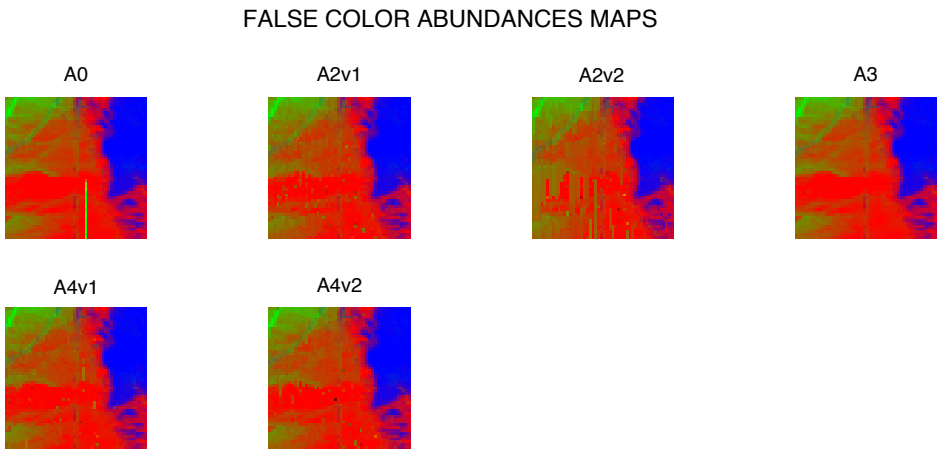


Figure 4.15: RGB abundances colormap of $R = 3$ endmembers of the Cuprite selection.

it does not converge in this experiments, due to the fact that the variability of the collected samples during the execution of the Metropolis-Gibbs step (see Algorithm 4) does not exhibit an asymptotic behavior. Instead, the algorithms A_2-A_4 come to convergence and provide valid estimation results.

In Figure 4.16 we can compare the algorithms performance through a scatterplot that relate RMSE with iterations. The RMSE obtained with the estimates of the algorithm A_6 is the lowest achieved, with a number of iterations smaller than A_0. The Algorithms A_2v2, A_3 and A_4 converge with a smaller number of iterations than A_0 and A_1, but the RMSE increases. Finally the algorithm A_2v1 provides the worst results in terms of RMSE, resulting inadecuate for the estimation task.

From the experiments reported in this subsection, we can conclude that the a approximation to the Gaussian distribution 2.3 with a linear distribution on the abundances estimation based on NCM, can improve the original algorithm. To conclude this section we remark that, besides the advantage of reducing both the iteration number and the RMSE values, the implemented linearized distribution can be computed through parallel operations, that involve each spectral band separately.

As demonstrated by our experiments with synthetic and real hyperspectral data, both techniques, applying the Genetic Algorithm sampling and the approximation of the Gaussian function, provides good results in different ex-

perimentas scenarios. The GAs prove their effectiveness if the experimental scene is characterized by a high noise level, while the linear approximation of the nonlinear distribution is effective when the SNR is included in the range 40-60 dB. The real hyperspectral image Cuprite, collected by the AVARIS are included in the beforementioned situation, as the measured noise level is about 40-44 dB.

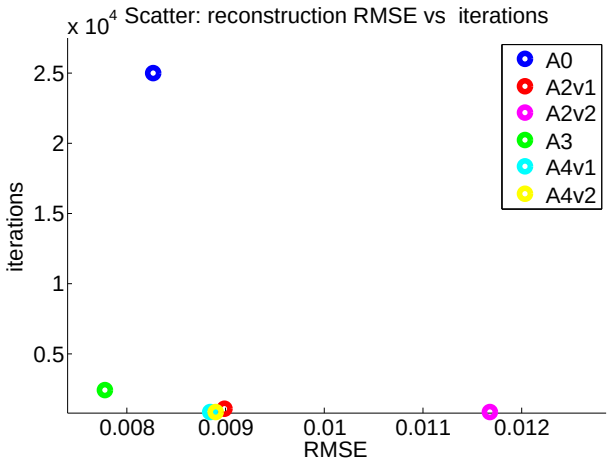


Figure 4.16: Scatter Plot of iterations and RMSE for Cuprite hyperspectral image.

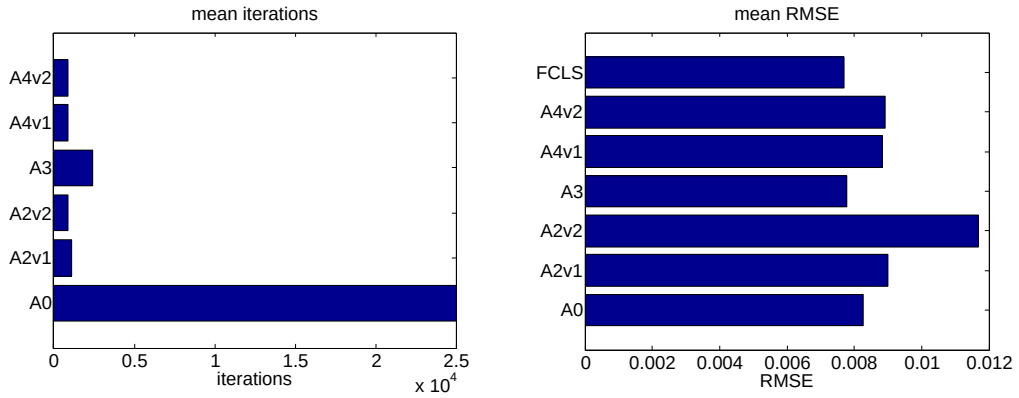


Figure 4.17: Bar diagram of the iteration number (left) and RMSE values (right), obtained from experiments with Cuprite hyperspectral image.

Bibliography

[Ahmed 2013] Ayman M. Ahmed, Mohamed El. Sharkawy and Salwa H. Elramly. *Hyperspectral bands prediction based on inter-band spectral correlation structure*. pages 86550Y–86550Y–14, 2013. (Cited on page 22.)

[Chang 1998] Chein-I Chang, Xiao-Li Zhao, M.L.G. Althouse and Jeng Jong Pan. *Least squares subspace projection approach to mixed pixel classification for hyperspectral images*. Geoscience and Remote Sensing, IEEE Transactions on, vol. 36, no. 3, pages 898–912, 1998. (Cited on page 22.)

[Dias 2012] J. Dias, A. Plaza, Dobigeon, Parente, Qian Du Du, Gader and J. Chanussot. *Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches*. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 5, no. 2, pages 354–370, April 2012. (Cited on page 20.)

[Dobigeon 2008] Nicolas Dobigeon, Jean-Yves Tourneret and Chein-I Chang. *Semi-Supervised Linear Spectral Unmixing Using a Hierarchical Bayesian Model for Hyperspectral Imagery*. IEEE Transactions on Signal Processing, vol. 56, no. 7-1, pages 2684–2695, 2008. (Cited on pages 26 and 27.)

[Dobigeon 2009] N. Dobigeon, S. Moussaoui, M. Coulon, J.-Y. Tourneret and A.O. Hero. *Joint Bayesian Endmember Extraction and Linear Unmixing for Hyperspectral Imagery*. Signal Processing, IEEE Transactions on, vol. 57, no. 11, pages 4355–4368, 2009. (Cited on page 27.)

[Eches 2010] O. Eches, N. Dobigeon, C. Mailhes and J.-Y. Tourneret. *Bayesian Estimation of Linear Mixtures Using the Normal Compositional Model. Application to Hyperspectral Imagery*. IEEE Signal Processing Magazine, vol. 19, no. 6, pages 1403–1413, 2010. (Cited on pages 21, 25, 27, 41, 42 and 67.)

[Gilks 1999] W. R. Gilks. Markov chain monte carlo in practice. Chapman and Hall/CRC, 1999. (Cited on pages 26 and 27.)

[González 2013] Carlos González, Sergio Sánchez, Abel Paz, Javier Resano, Daniel Mozos and Antonio Plaza. *Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing*. Integr. VLSI J., vol. 46, no. 2, pages 89–103, March 2013. (Cited on page 41.)

[Harsanyi 1994] J.C. Harsanyi and Chein-I Chang. *Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach*. Geoscience and Remote Sensing, IEEE Transactions on, vol. 32, no. 4, pages 779–785, 1994. (Cited on page 16.)

[Johnson 1993] D.G. Johnson and Loughborough Univ. Business School (GB). The triangular distribution as a proxy for the beta distribution in risk analysis. 1993. (Cited on page 57.)

[Keshava 2002] N. Keshava and J. Mustard. *Spectral unmixing*. IEEE Signal Processing Magazine, vol. 5, pages 44–56, Jan. 2002. (Cited on page 19.)

[Li 2004] Jiang Li. *Wavelet-based feature extraction for improved endmember abundance estimation in linear unmixing of hyperspectral signals*. Geoscience and Remote Sensing, IEEE Transactions on, vol. 42, no. 3, pages 644–649, 2004. (Cited on page 22.)

[Manolakis 2003] Dimitris Manolakis, David Marden and Gary A. Shaw. *Hyperspectral Image Processing for Automatic Target Detection Applications*, 2003. (Cited on page 1.)

[Nascimento 2004] Jose M. P. Nascimento and Jose M. B. Dias. *Vertex Component Analysis: A Fast Algorithm to Unmix Hyperspectral Data*. IEEE TRANS. GEOSCI. REM. SENS, vol. 43, pages 898–910, 2004. (Cited on pages 16 and 20.)

[Peg Shippert 2003] Inc. Peg Shippert Ph.D. Earth Science Applications Specialist. Research Systems. *Introduction to Hyperspectral Image Analysis*, 2003. Available on line. (Cited on page 1.)

[Robert 2005] Christian P. Robert and George Casella. Monte carlo statistical methods (springer texts in statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. (Cited on page 30.)

[Scott 1979] David W. Scott. *On optimal and data-based histograms*. Biometrika, vol. 66, no. 3, pages 605–610, 1979. (Cited on page 46.)

[Smith 2012] Randall B. Smith. *Introduction to Hyperspectral Imaging*, 2012. Available on line. (Cited on page 1.)

[USGS 2012] USGS. *USGS Libraries*, 2012. (Cited on pages 69 and 70.)

[Veit] Thomas Veit, Jérôme Idier and Saïd Moussaoui. *Rééchantillonnage de l'échelle dans les algorithmes MCMC pour les problémes inverses bilinéaires*. Traitement du Signal. (Cited on page 22.)

[Winter 1999] Michael E. Winter. *N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data.* pages 266–275, 1999. (Cited on page 15.)