Image Analysis Based on Tensor Representations

Mohamad Jouni

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE GRENOBLE ALPES

Spécialité : **SIGNAL IMAGE PAROLE TELECOMS**

Arrêté ministériel : 25 mai 2016

Présentée par

Mohamad JOUNI

Thèse dirigée par **Pierre COMON,** Université Grenoble Alpes,
et codirigée par **Mauro DALLA MURA,** Université Grenoble Alpes

préparée au sein du **Laboratoire Grenoble Images Paroles Signal Automatique (GIPSA-Lab)**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Analyse d'image basée sur les représentations de tensors

Image Analysis Based on Tensor Representations

Thèse soutenue publiquement le **29 janvier 2021**,
devant le jury composé de :

Monsieur Nicolas LE BIHAN
Directeur de recherche, CNRS délégation Alpes, Président du jury
Monsieur Cédric FEVOTTE
Directeur de recherche, CNRS délégation Occitanie Ouest, Rapporteur
Monsieur Paul SCHEUNDERS
Professeur, Universiteit Antwerpen, Rapporteur
Monsieur Kuniaki UTO
Professeur assistant, Tokyo Institute of Technology - Japon, Membre
Monsieur Naoto YOKOYA
Professeur associe, RIKEN - Wako - Japon, Membre
Monsieur Lucas DRUMETZ
Maitre de conférence, IMT Atlantique, Membre
Monsieur Pierre COMON
Directeur de recherche, CNRS délégation Alpes, Directeur de thèse

UNIVERSITÉ DE GRENOBLE ALPES

ED EEATS

l'École Doctorale Électronique, Électrotechnique, Automatique, Traitement du Signal

T H E S I S

submitted for the degree of

Doctor of Philosophy

from UNIVERSITÉ GRENOBLE ALPES

Specialty : SIGNAL IMAGE PAROLE TELECOMS

Presented and defended

Mohamad JOUNI

Image Analysis Based on Tensor Representations

Thesis directed by Pierre COMON
and co-directed by Mauro DALLA MURA

Prepared at GIPSA-Lab lab
(Grenoble Images Parole Signal Automatique)

defended 29/01/2021

In front of the jury composed by :

President of Jury :	Nicolas LE BIHAN	-	GIPSA-Lab, Grenoble, France
Reviewers :	Cedric FÉVOTTE	-	CNRS, IRIT, Toulouse, France
	Paul SCHEUNDERS	-	Vision Lab, Antwerp, Belgium
Examiners :	Kuniaki UTO	-	Tokyo Tech, Tokyo, Japan
	Naoto YOKOYA	-	RIKEN, Tokyo, Japan
	Lucas DRUMETZ	-	IMT Atlantique, Brest, France
Director :	Pierre COMON	-	GIPSA-Lab, Grenoble, France
Co-director :	Mauro DALLA MURA	-	GIPSA-Lab, Grenoble, France

Acknowledgements

The last four years, including my masters studies, have been full with continuous support from my surrounding, which includes my professors, family, dear friends, colleagues, and co-workers.

During my masters, I met a lot of amazing people, especially my two professors and eventually my internship and PhD supervisors, Prof. Pierre Comon and Prof. Mauro Dalla Mura, for whom no word can express my gratitude and respect. For both my internship and PhD thesis, I thank both of them for their continuous advice, the rich scientific background and discussions, and for believing in me and having to put with me through good and bad times. It has truly been a blessing meeting them not only as thesis supervisors, but also as instructors and advisors. None of this would have been possible without them.

Throughout my academic journey so far, my family has been my biggest supporter. To my parents, siblings, and lover Aya, who have been cheering me on and following my progress with all its checkpoints, this report is mostly dedicated to you.

This goes without mentioning GIPSA-Lab (Grenoble Images Parole Signal Automatique Laboratoire) in Grenoble, France and all of its kind staff, some of whom were my professors during my masters studies. I spent four years as part of GIPSA-Lab, studying, working, connecting with people, and joining a lot of its scientific and social events (seminars, celebrations, bowling, 2018 FIFA World Cup, GIPSA-Days, etc), and it was all such a great experience.

I would like to thank Prof. Kuniaki Uto and Shinoda-Lab, from Tokyo Institute of Technology in Japan, for kindly hosting me for a 3-months research visiting fellowship with Prof. Dalla Mura and Dr. Daniele Picone. This short visit, with the help of Prof. Uto, gave me firsthand experience in image acquisition and processing through drones and tools related to my thesis, and allowed me to meet a lot of researchers in related fields (by visiting different labs and joining seminars and a Summer School program) and gain a wider insight on different applications and skills that are now part of my scientific interest.

Very special thanks to the members of my CSI (Comité Suivi de Thèse) committee, Prof. Nicolas Le Bihan and Prof. Simon Barthelmé, as well as the members of the PhD committee for reviewing this work, for their interest, and for making all of this journey possible.

I would also like to mention my dear friends and colleagues for all of their direct and indirect support and motivation throughout the years, whether scientific or not. This includes my friend Iyad Tout, who came to Grenoble and physically helped me setup my defense. After all, I strongly believe that no such journey is worth it without sharing it with friends.

Finally, I would like to mention two works of fiction, Naruto and One Piece (by the authors Masashi Kishimoto and Eiichiro Oda respectively), that have been great sources of motivation ever since I was little.

Contents

List of Figures

List of Tables

Table of acronyms

RGB	Red Green Blue
SAR	Synthetic-Aperture Radar
2D	Two-Dimensional
N-D	N-Dimensional
MSI	MultiSpectral Images
HSI	HyperSpectral Images
MM	Mathematical Morphology
HOSVD	High-Order Singular Value Decomposition
CP	Canonical Polyadic
ALS	Alternating Least Squares
AO-ADMM	Alternating Optimization - Alternating Direction Method of Multipliers
ASC	Abundance Sum-to-one Constraints
UAV	Unmanned Aerial Vehicles
DS4G	Data Science for Geoscience
SVD	Singular Value Decomposition
PCA	Principal Component Analysis
PC	Principal Component
NMF	Nonnegative Matrix Factorization
BSS	Blind Source Separation
BTD	Block Term Decomposition
TT	Tensor Train
TR	Tensor Ring
HSI-CSR	Hyperspectral Image Compressive Sensing Reconstruction
LS	Least Squares
NCP	Nonnegative Canonical Polyadic

PARALIND	PARAllel profiles with LINear Dependencies
CONFAC	CONstrained FACtors
NALS	Nonnegative Alternating Least Squares
ProCo ALS	Projected-Compressed Alternating Least Squares
AO-PDS	Alternating Optimization - Primal Dual Splitting
DR	Dimensionality Reduction
MP	Morphological Profiles
AP	Attribute Profiles
EMP	Extended Morphological Profiles
EAP	Extended Attribute Profiles
EMAP	Extended Multi-Attribute Profiles
TPCA	Tensor Principal Component Analysis
AMD	Additive Morphological Decomposition
SU	Spectral Unmixing
LMM	Linear Mixing Model
CNMF	Coupled Nonnegative Matrix Factorization
ELMM	Extended Linear Mixing Model
SRI	Super-Resolution Image
RegELMM	Regulated Extended Linear Mixing Model
CPD	Canonical Polyadic Decomposition
DMP	Differential Morphological Profile
SE	Structuring Element
DFC	Data Fusion Contest
SVM	Support Vector Machine
OA	Overall Accuracy
AA	Average Accuracy
Rec. Error	Reconstruction Error
NAMD	Nonnegative Additive Morphological Decomposition

HU	Hyperspectral Unmixing
CLS	Constrained Least Squares
NCPD	Nonnegative Canonical Polyadic Decomposition
RMSE	Root Mean Squared Error
SAD	Spectral Angular Distance
GT	GroundTruth
FNN	Feedforward Neural Network

Chapter 1

Introduction

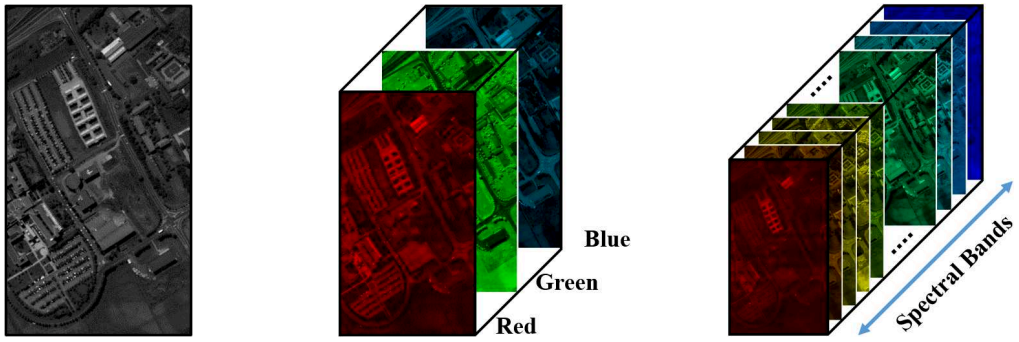
Sommaire

"Image Analysis Based on Tensor Representations": On one hand, "image analysis" represents a wide area of applications. On another, "tensor representations" refers to a whole mathematical field called Multi-linear Algebra (or Tensor Algebra). In an attempt to reduce the gap between the two disciplines, I dedicate this general and short introduction to anyone who is interested, hoping it serves as a smooth and easy way into the thesis.

1.1 Short Overview

In general, *image analysis* represents a wide area of applications such as Computer Vision, Computational Imaging, Remote Sensing, Biomedical Engineering, etc. It concerns any application that requires the study of images or image representations of the acquired data. Like any form of signal processing, image processing relies on many mathematical foundations such as statistics, algebra, topology, mathematical morphology, etc, which allows to explore the features present in the data that come from the real world and make sense out of them. Images can contain a lot of information that, intuitively speaking, may appear to lie visually in a two-dimensional plane (e.g. horizontal and vertical spatial positioning of the pixels of a scene). However, digitally, such information can be stored in different forms depending on the type of *image* and *application*. For example, an image can be grayscale, color RGB (Red, Green, Blue), hyperspectral, a set of X-rays, Synthetic-aperture radar (SAR), etc; and an application can be object detection, scene classification, blind source separation, tomography, seismic imaging, etc.

To give some examples, as illustrated in Figure 1.1, a grayscale image can be digitally stored as a two-dimensional (2D) array, a color RGB image can be stored as a 3D array con-



(a) Grayscale (b) RGB (c) Hyperspectral

Figure 1.1: Examples of image types using grayscale samples taken from the HSI of Pavia University. (The colors are only for display purposes.)

sisting of three grayscale images for the three colors, and Hyperspectral Images (HSI) can be stored as a 3D array consisting of hundreds of grayscale images spanning an almost continuous range of frequencies. Moreover, images can be acquired over different time stamps or any parametric diversities, allowing to extend their storage to probably *higher-dimensional arrays*. Processing such data sets and extracting meaningful features from them require suitable representations with mathematical foundations, which can be found in statistics, inverse problems, algebra, etc. For many applications, especially in classical approaches, we find that such multi-dimensional data arrays are either (a) considered as a collection of 2D arrays that would be processed independently, or (b) entirely reshaped as 2D arrays in order to fit with matrix representations, allowing access to *linear algebra* and *matrix decomposition* techniques. For example, as illustrated in Figure 1.2, a 3D HSI can be reshaped into a 2D array by rearranging the pixels in a lexicographic order (i.e. the grayscale image slices are vectorized), which can be represented as a matrix where pixels are shown as rows of spectral information. Then, pixels are seen as samples, and spectral bands as features, which can for instance be exploited through statistical models.

However, we find that matrix modeling is insufficient in other, recent applications, and more specifically, where *high-order*[1] processing and decomposition of the data is recommended. For instance, there can be cases where: (a) the spatial information between the two modes of pixels is very important that rearranging the modes of pixels becomes a problem [53, 108], and (b) the images are collected or represented with multiple diversities [102, 103, 56] where the diversities may be defined by: (i) natural acquisition (such as multi-temporal or multi-angular HSI data) [102], (ii) artificially extracted features (such as HSI with neighborhood patches or Mathematical Morphology) [103, 56], (iii) a combination of them, etc. With such data sets, especially those with multiple diversities, it is better to think of each diversity as a subspace of its own. However, as mentioned earlier, classical attempts to process high-order data using matrix representations include (a) merging the modes of pixels and diversities (as

[1]The number of ways of an array refers to the number of its indices, which is sometimes referred to as its *order* or the number of its *modes*. For example, a HSI of dimensions $I_1 \times I_2 \times J$ is said to have three *ways* or *modes*, and is called a *three-way array* or a *data set of order three*. Data sets with order 3 or above are described as *multi-way*, *multi-modal*, or *high-order* tensors.

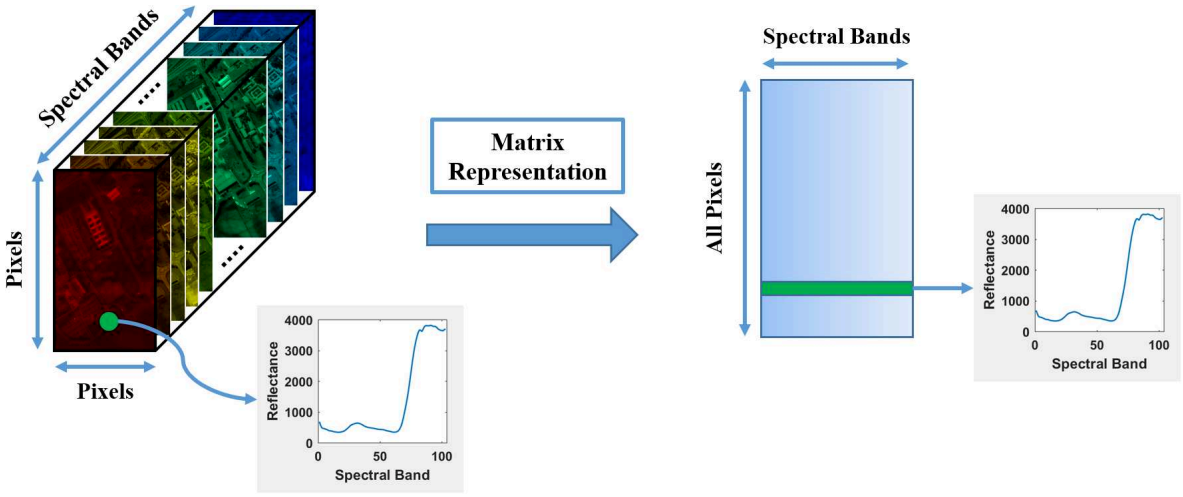


Figure 1.2: Reshaping a 3D array hyperspectral image (the set of 2D image slices on the left) into a 2D array such that the pixels are rearranged into a lexicographic order (on the right). Each row in the matrix representation is a spectral reflectance vector.

illustrated in Figure 1.3 for a HSI with a temporal diversity) or (b) processing each of the diversities independently to fit for matrix techniques. The problem is that the former destroys the natural ordering of the data and the algebraic subspace entity of each mode and diversity, and the latter ignores the complex inter-modal relationships. In order to avoid these scenarios, we need to rely on *high-order*, *tensor representations*.

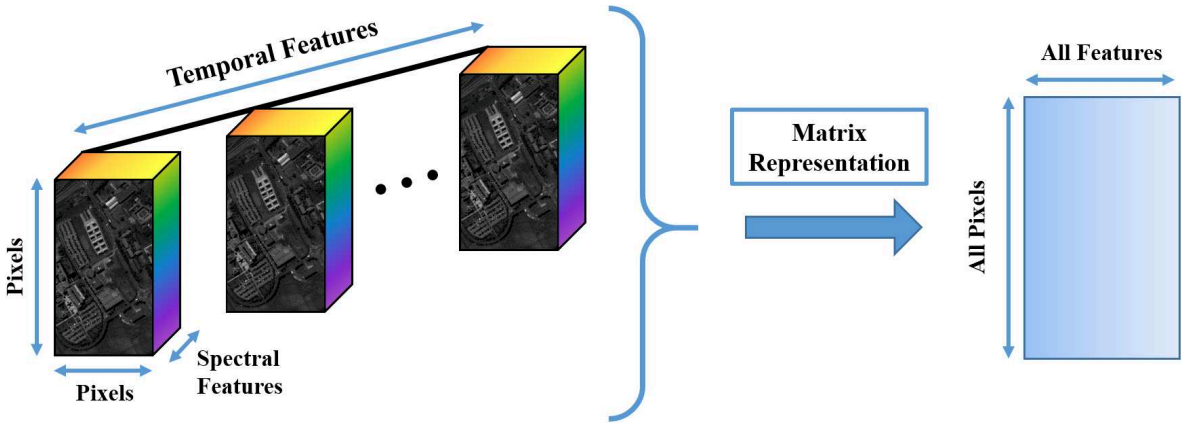


Figure 1.3: Matricization of a fourth-order HSI tensor in time series (represented as a set of third-order HSI blocks on the left) into a 2D array such that the pixels are rearranged into a lexicographic order along the first mode, and the spectral and temporal features are reordered along the second mode (on the right). Each row in the matrix representation is a vector of mixed spectral and temporal properties.

Almost analogously to how matrices and their tools (e.g. operators, ranks, factorization, etc) are defined in linear algebra, *tensors* and their tools extend those towards *multi-linear algebra*. Tensors are high-order data structures that can be represented as multi-dimensional

arrays of data. As such, a first-order tensor is just a vector, and a second-order tensor is a matrix. Tensor analysis [29] of the data has been applied in a variety of applications ranging from chemometrics to psychometrics, biomedical signal processing, neural networks, social networks, etc [65, 25, 29]. Recently, tensor analysis is being integrated in image processing applications, so their tools are used to solve many problems related to high-order image analysis. In Chapter 1, we compile in some details the basics of tensors and some of their applications for remote sensing images.

Now, what does this mean in terms of acquired image data? During the past couple of years, the data that I worked with were strictly image arrays of order ≥ 3, meaning that I have been dealing with not only the relationships between a pixel and its features, but also those between the features themselves (e.g. between spectral and temporal features). For instance, strictly speaking, image data arrays are formed of *numbers*. However, these numbers are acquired from nature and everyday-activities based on physical and meaningful stimulus coming from different sources (as mentioned earlier), so they have *patterns* waiting to be extracted and made sense of. Here, we would like to describe the utility of tensors for multi-modal image analysis in three general points:

- Due to the increasing availability of high-order and big image data [26], it is becoming increasingly important to rely on tensor representations and observe these patterns in their raw, high-order form with their inter-modal relationships. Mathematically speaking, tensor algebra is able to jointly show the correspondences between the subspaces that form a multi-modal image tensor through *tensor decomposition*.
- Tensor decomposition, in all its different forms and definitions, is the most important tool to process tensor data for feature extraction. Since such tools are abstractly defined in multi-linear algebra with available algorithms, it is important to explore their utility in image analysis and extend their models to possible *physical conditions*. Since tensor decomposition respects the ordering of each of the modes, it is more flexible to impose diversity-dependent regularization terms and adapt to human intuition than the case of matrix factorization.
- Moreover, since images are visual in nature, and since tensors are pretty abstract, expectations and interpretations of the results heavily depend on both. It is hard to talk about one aspect without the other, and here lies the beauty of this harmony between the two worlds, but also the difficulty to explain it shortly in words. For that, in Section 1.2, we show a simple example of decomposing a synthetic image tensor.

In other words, the goal is to *jointly* explore high-order relationships between the diversities as tensor decomposition techniques can reveal interesting patterns with inter-modal and subspace relationships, which would be lost through matrix representations. Additionally, tensor decomposition has an important advantage when it comes to the identifiability and uniqueness of the solution, all of which we discuss in more details in Chapter 2. Now that a rather broad link is developed between image analysis and tensor decomposition, and since time does not allow to practically explore all the different areas, we briefly expand on the elements, applications, and goals that were treated in the scope of this thesis.

1.1.1 Tensor Decomposition of Multi-featured HSIs

In terms of image analysis, the focus is going to be on remote sensing images and applications. In other words, the types of images of this thesis are going to be HSIs of landscapes obtained by remote sensing platforms for Earth observation. It is good to note, though, that while the focus is on remote sensing images, the issues and methods that are tackled apply also to HSIs of any other field of research, like biomedical imaging for instance. Regarding HSIs, to begin with, they are natural three-way blocks of data, which already allows access to tensor tools, and such related works can be found in [61, 108, 79] where the HSIs are processed in their three-dimensional array form using tensor decomposition. In this thesis however, we consider HSIs rather from the perspective of each single pixel than the data set as a whole. As such, each pixel in a HSI is a *sample* that represents a vector of spectral *features*.

At this stage, one might wonder: Doesn't this way of looking at HSIs suggest that the three-way data at hand can be simply rearranged as a matrix (as indicated in Figure 1.2)? And then there would not be a need to further complicate things in the scope of a full thesis? Indeed, that can be an option (regardless of its limitations; e.g., solution uniqueness). Nevertheless, as described in earlier paragraphs, HSIs can also be represented with different types of features, which we will sometimes coin as being *"multi-feature"*-ed, and which is exactly the type of data that we target in our applications. As multi-featured data are becoming highly accessible, from a signal processing point of view, it would be better to start widely relying on the more direct and fitting tensor analysis, which this thesis is dedicated for.

Regarding the nature of additional features, in general, our main interest goes towards those that are *extracted* from the HSI itself rather than the ones that are naturally acquired. In fact, it is a standard practice in remote sensing, and in general any type of image processing actually, to use or extract spatial features from images and consider them for image analysis, which can be for instance to detect or discover certain spatial features or patterns (e.g., scene classification [50], spectral unmixing [107], texture analysis [5]). After all, images themselves are units of visual and spatial information that *should* be exploited. The approaches may differ as to how these features are modeled or processed, but we focus here on representing them within tensors. Specifically, we exploit the area of *Mathematical Morphology* (MM) [85, 86, 73], which is a set of image-to-image transformations that extract spatial features from images based on the size of spatial objects and their brightness levels, among others (see Section 4.2 for some details on morphological operators and properties). Exploring such features comes with potential advantages such as:

- Acquisition: Compared to natural multi-featured HSIs such as time-series, it requires us to acquire only one instance of a HSI of the scene. (That is, of course, when we are not primarily interested in any natural evolution of the physical materials of the scene.)
- Uniqueness: Compared to matrix factorization techniques, we can have a unique decomposition of the data under mild conditions. Moreover, we are able to observe multiple features at once for each extracted material, and not only spectral ones. In case of MM, this also means that we extract or span a morphological subspace for the data.

- Low-rank Approximation: Compared to analysing HSIs in their original third-order form, rearranging the modes of pixels allows to focus on the pixel-features relationship and significantly drops the rank of the data. We note that as pointed out in Chapter 5, even though the spatial structure is ignored when the pixels are rearranged, it is still possible to impose predefined neighborhood-local spatial structures such as the case of patches and MM [104, 58, 56, 57]. In other words, it can only suffice to use small neighborhoods as additional spatial information instead of the spatial structure of the whole image which would be very high-rank.

That said, many of the tensor interpretations given later for applications with MM are general and can also apply to other types of additional features.

Among the many applications in the world of hyperspectral imaging, we explore those that are related to compression, scene classification, and spectral unmixing. In each of these applications, instead of the classical matrix-based approaches that disregard the multi-modal properties, we use tensor modeling, which means that we also explore the theoretical, applicative, and algorithmic aspects of tensor analysis. This includes:

- Abstract definition of tensors and their operators.
- Different decomposition techniques such as High-Order Singular Value Decomposition (HOSVD) and Canonical Polyadic (CP) Decomposition.
- Conditions of uniqueness that are possibly encountered in image analysis problems.
- Imposing nonnegativity, sparsity, and simplex constraints.
- Algorithms that were used such as Alternating Least Squares (ALS) and Alternating Optimization Alternating Direction Method of Multipliers (AO-ADMM).

With that, we are ready now to show a simple synthetic image example of how tensor decomposition can work with multi-modal images.

1.2 Synthetic Image Tensor

Here, we present a simple example of decomposing a synthetic RGB image tensor where the objects of the scene are fading or increasing in appearance (color). The point is to see an example of how the image-based tensor data can be reflected through tensor decomposition. Looking at Figure 1.4, we have images of size 36×100 containing three squares: red, green, and blue, observed over 7 time stamps. The red square is fading over time, the green one is constant, while the blue one increases in appearance. Moreover, this is an RGB image, so we have 3 spectral bands representing the red, green, and blue wavelengths. This is a fourth-order tensor of dimensions $36 \times 100 \times 3 \times 7$, but we reshape it into a third-order tensor of dimensions $3600 \times 3 \times 7$ (Figure 1.5) for decomposition.

Figure 1.4: RGB image slices showing the color evolution of the squares

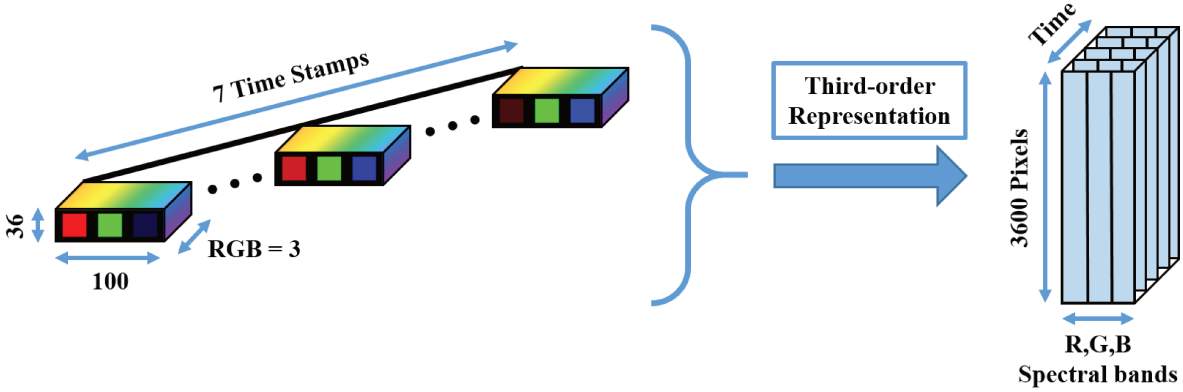


Figure 1.5: Reordering the pixels of the spectral-temporal HSI in a lexicographic order

Using tensor CP decomposition, which is one form of tensor decomposition (see Section 2.4.2), it is possible to break down the tensor's third-order structure into three matrices $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$, each describing or showing the patterns of one of the modes {Pixels, RGB bands, Time} respectively. In particular, the columns of $\boldsymbol{A}$ can contain factorized spatial objects whose spectral and temporal features are described by the columns of $\boldsymbol{B}$ and $\boldsymbol{C}$ respectively, which should show RGB color patterns and reflect their temporal changes respectively.

First, we note that each column of $\boldsymbol{A}$ is reshaped into a 36×100 grayscale image, shown in Figure 1.6. We notice that each image contains one of the squares separately.

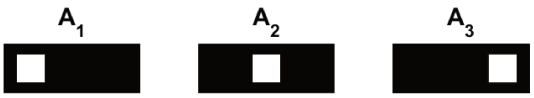


Figure 1.6: The columns $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3\}$ of $\boldsymbol{A}$, reshaped and viewed as grayscale images

Then, we look at the (normalized) columns of $\boldsymbol{B}$ and $\boldsymbol{C}$, noting that in CP decomposition, a column $\boldsymbol{A}_r$ corresponds to columns $\boldsymbol{B}_r$ and $\boldsymbol{C}_r \; \forall \; r = \{1, 2, 3\}$:

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} <=> \begin{pmatrix} R \\ G \\ B \end{pmatrix} ; \; \boldsymbol{C} = \begin{bmatrix} 0.62 & 0.37 & 0.00 \\ 0.52 & 0.37 & 0.10 \\ 0.42 & 0.37 & 0.20 \\ 0.31 & 0.37 & 0.31 \\ 0.20 & 0.37 & 0.42 \\ 0.10 & 0.37 & 0.52 \\ 0.00 & 0.37 & 0.62 \end{bmatrix} <=> \begin{pmatrix} t = 1 \\ t = 2 \\ t = 3 \\ t = 4 \\ t = 5 \\ t = 6 \\ t = 7 \end{pmatrix}$$

According to $\boldsymbol{B}$, the first column indicates *red* color, which describes the RGB signature of the square extracted in $\boldsymbol{A}_1$, the second indicates *green*, which describes that of $\boldsymbol{A}_2$, and the third indicates *blue*, which describes that of $\boldsymbol{A}_3$. As for $\boldsymbol{C}$, the first column shows decreasing values

from $t = 1$ to $t = 7$, which describes the fading of the red square, the second column remains constant, which describes the stillness of the green square, while the third shows increasing values, which describes the gradual appearance of the blue square.

The presented synthetic example shows how intuitive tensor decomposition can be with multi-modal data, but it is a very simple and direct case. In fact, there is much more to matrix and tensor decomposition in terms of techniques and conditions, adding to them that real-life images contain more complicated features (we refer to [102] for an application of real time-HSI tensors). That said, image analysis presents us with a couple of challenges when it comes to tensor modeling, especially with real HSI tensors, and we briefly describe some of those that were tackled during the thesis.

1.2.1 Thesis Challenges

To begin with, one goal of this thesis is to present the common literature between tensor decomposition and image analysis. It is a challenge of itself to find a unified compilation for the challenges and issues of tensor decomposition that arise in image analysis applications. One would usually have to go through the numerous individual papers in the literature, many of which deal purely with issues related to either one of the two disciplines, and few of which deal with both. For that, Chapter 2 represents such compilation that goes in details with the parts that are directly relevant to the works of this thesis (i.e., Chapters 3, 4, and 5) and provides many references for those that are potentially useful in image analysis but are not tackled in the scope of this thesis.

Furthermore, we find that there are algorithmic and application-oriented challenges in the literature. In the pure algorithmic sense, we still find problems that require special modeling and constraints. For example, as HSIs can be seen as very big data sets, and especially in the presence of constraints, computing their CP decomposition can be demanding and requires compression. An algorithm has already been proposed for fast computations of such cases [27], but it poses some computational problems when it comes to constraints that rely on hard-thresholding, for which we propose a greedy fix in Chapter 3 [60]. Moreover, another algorithm has been proposed for flexible modeling of the constraints in CP decomposition [53], which relies on breaking down the non-convex inverse problem into alternating convex ADMM sub-problems. In Chapters 4 [56, 58, 57] and 5 we rely on AO-ADMM for multi-feature hyperspectral scene classification and unmixing respectively, in which we respectively propose to use compression (Section 4.3.1) and Abundance Sum-to-one Constraints (ASC) (Section 5.3) through AO-ADMM.

On the application-side, as we tackle scene classification and spectral unmixing through CP decomposition, we view the problems from the perspective of multi-linear algebra and discuss some challenges related to the smooth transitioning from classical matrix approaches to tensor ones, choosing a value for the tensor rank before applying decomposition (which is also the number of extracted materials), expecting or physically interpreting the results of the decomposition in a multi-linear sense, and properly explaining and presenting the ideas

and contributions because usually, with tensor representations, a lot of fields and ideas are inter-connected and move in parallel.

1.3 Manuscript Overview

This manuscript starts with an overview of matrix and tensor decomposition and some of their applications in remote sensing image analysis. After that, we present the findings of this thesis. Regarding the contributions, application-wise, the focus was on HSIs and *nonnegative* tensor decomposition, with a highlight on the usefulness of building a third diversity of features using MM.

That said, the rest of the manuscript is structured as follows. In Chapter 2, we start with the mathematical aspects of matrix and tensor decomposition that are important for image analysis. This includes definitions, basic operators, decomposition techniques, uniqueness conditions, and algorithms. At the end, we talk about some applications of remote sensing image analysis where tensor representations were involved. After the literature overview, in Chapter 3, we present our first contribution, which is rather algorithmic and concerns the computation of nonnegative tensor decomposition, noting that nonnegativity is an important constraint for image analysis. In Chapter 4, we present our second, third, and fourth contributions, which concern hyperspectral image scene classification using Mathematical Morphology and tensor CP decomposition. In Chapter 5, we present our fifth contribution, which concerns hyperspectral multi-feature unmixing using tensor CP decomposition, and can be seen as a natural extension to Chapter 4. Following the challenge of reasoning on the choice of the tensor rank in Chapter 5, we explore in a current work (i.e., it is currently under development) the low-rank tensor representations of tensor data that are built from sequential filtering of the original HSI such as the case of MM. Finally, we draw out some conclusions and talk about different challenges and potential perspectives.

1.3.1 Scientific Contributions

So far, the published works of the PhD are divided into three conference papers [60, 56, 58], which were presented by me in each of the conferences, and one journal paper [57]. The first conference paper [60] deals with the algorithmic contribution and is presented in Chapter 3, while the others and the journal [56, 58, 57] deal with HSI scene classification using MM and CP decomposition and are presented in Chapter 4. That said, materials of Chapter 5 are planned for further development and publishing in the future.

Beside the published works, during this PhD, we were hosted at Tokyo Institute of Technology - School of Computing, Japan, for a research collaboration on a project funded by JST Presto [98]: “Estimation of crop vitality based on Tensor decomposition and data fusion of multi-modal, multi-temporal leaf-scale aerial images”, under the direction of Prof. Kuniaki Uto, where tensor decomposition of aerial images of crops is proposed in order to control the

health of the crops under physical conditions such as sunlight and color of the leaves. During this stay in Japan, we were partially involved in the problem statement, and we conducted two field campaigns and acquired real-life RGB, multi-spectral and hyper-spectral images of the rice fields of Tsuruoka, Japan, using Unmanned Aerial Vehicles (UAV)-mounted RGB, multi-spectral and hyper-spectral cameras.

Moreover, we prepared a course[2] on feature extraction with matrix and tensor techniques for the doctoral school "Data Science for Geoscience (DS4G) 2020" by using notebooks[3] prepared with the tensor library of TensorLy (built in Python) [66].

[2]https://github.com/DataScience4Geoscience/Toulouse2020/tree/master/Courses/Feature%20Extraction
[3]https://github.com/DataScience4Geoscience/Toulouse2020/tree/master/Notebooks/Feature%20Extraction

Chapter 2

Literature Review

Sommaire

2.1 Introduction

One problem that I encountered while working on my thesis was managing the huge piling of papers that I ended up marking as important references for the various issues and findings in tensor algebra and image analysis, whether in theory or applications, that I needed to regularly refer to for my work. In other words, while it is fortunate to be able to find tutorials and answers to most of the problems that arise while working on tensor analysis of high-order image data, these answers would be available but spread over numerous papers, and thus it was a bit hard and time-consuming to keep track of them as well as refer to them on a regular basis. Additionally, seeing as the tensor analysis of high-order image data has recently been

rising in popularity that it has been given its own sections in some conferences (such as in IGARSS 2019), it would be of great importance to have a summary of the latest corresponding developments and concerns along with references and examples of image applications from the literature. As such, this chapter serves as a structured compilation or documentation that deals with the common problems of the two major disciplines, i.e., tensor decomposition and image analysis. It represents a theoretical and applicative account of the different aspects that are relevant to these elements and will be recalled often throughout the following chapters.

Starting with the theoretical part, in Section 2.2, we start with an abstract definition of a tensor, of which a matrix is a special case, then we describe some mathematical operators that sit at the core of decomposing and reconstructing tensors. After that, in Sections 2.3 and 2.4, we talk about matrix and tensor decomposition respectively. Here, we note that since tensor decomposition is often seen as the multi-linear extension of matrix factorization, and since passing by the matrix case can make the tensor one easier and more intuitive to explain, we split between the two cases. In particular, in Section 2.3, we talk about some of the most popular matrix factorization techniques and describe their properties, then, in Section 2.4, we move to tensor decomposition and present it in details, which includes the different techniques that are relevant to image analysis, their properties, cost functions, constrained versions, uniqueness conditions, and some algorithmic aspects. Finally, in Section 2.5, we go through some of the applications in image analysis as examples of how matrix and tensor decomposition is used in this context.

2.2 Definitions

Aside from their representations as multi-dimensional arrays, tensors are mathematical objects defined in Algebra at the base of vector spaces and mathematical operators (see [29, 49, 68] for detailed descriptions and definitions), which gives them a solid importance in data processing. As such, this section serves as a brief mathematical description of tensors and some basic operators.

First, we describe the *outer product* $\otimes$ of vectors and how it contributes to the formation of a tensor subspace, i.e., through the outer product of multiple vector subspaces, which in turns sets the mathematical and physical significance of tensor decomposition. Tensors are originally defined independently of any system of coordinates. But once a coordinate system is fixed, they are defined by a table of numbers containing their coordinates. We shall restrict our approach to such a context, and treat tensors as mere tables of numbers. Second, we describe some basic operators that are directly or indirectly used in tensor tools. Application-wise, images are acquired and represented as collections of real numbers, so all of the following algebraic definitions and operators are defined by default in the field of real numbers $\mathbb{R}$.

2.2.1 Outer Products and Tensors

The *Outer Product* of two vectors $\boldsymbol{a} \in \mathbb{R}^I$ and $\boldsymbol{b} \in \mathbb{R}^J$ results in a matrix $\boldsymbol{T} \in \mathbb{R}^{I \times J}$ as follows:

$$\boldsymbol{T} = \boldsymbol{a} \otimes \boldsymbol{b} \rightarrow t_{i,j} = a_i b_j$$

$\forall\, i \in \{1, \ldots, I\}$ and $\forall\, j \in \{1, \ldots, J\}$. Moreover, the *Outer Product* of three vectors $\boldsymbol{a} \in \mathbb{R}^I$, $\boldsymbol{b} \in \mathbb{R}^J$ and $\boldsymbol{c} \in \mathbb{R}^K$ results in a third-order tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ as follows:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c} \rightarrow t_{i,j,k} = a_i b_j c_k$$

$\forall\, i \in \{1, \ldots, I\}$, $\forall\, j \in \{1, \ldots, J\}$, and $\forall\, k \in \{1, \ldots, K\}$. A vector and a matrix can then be seen as first-order and second-order tensors respectively.

More generally, the *Outer Product* of N vectors $\{\boldsymbol{h}^{(1)} \in \mathbb{R}^{I_1}, \ldots, \boldsymbol{h}^{(N)} \in \mathbb{R}^{I_N}\}$ results in an N-th order tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ as follows:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{h}^{(1)} \otimes \ldots \otimes \boldsymbol{h}^{(N)} \rightarrow t_{i_1, \ldots, i_N} = h^{(1)}_{i_1} \ldots h^{(N)}_{i_N}$$

$\forall\, i_d \in \{1, \ldots, I_d\}$ and $\forall\, d \in \{1, \ldots, N\}$.

In fact, a *real tensor* is an element of $\mathbb{R}^{I_1} \otimes \ldots \otimes \mathbb{R}^{I_N}$ (or $\mathbb{R}^{I_1 \times \ldots \times I_N}$ for short). Consequently, a nonzero tensor is expressed as the sum of outer products of nonzero vectors, and the minimal number of summands for the following representation to be exact represents the *rank* of the tensor:

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \boldsymbol{h}^{(1)}_r \otimes \ldots \otimes \boldsymbol{h}^{(N)}_r \tag{2.1}$$

where R would be the tensor rank and $\boldsymbol{h}^{(d)}_r \in \mathbb{R}^{I_d}$, $\forall\, r \in \{1, \ldots, R\}$ and $\forall\, d \in \{1, \ldots, N\}$. Expression 2.1 sets the case of Canonical Polyadic (CP) decomposition [29], which we talk about in more details in Section 2.4.2. Besides, in Section 2.4, we see other forms of tensor decomposition, some of which are defined in the framework of linear subspace operators between tensors and matrices (see the contraction operator in expression (2.2) for instance), and others of which are derived as special cases of CP decomposition.

2.2.2 Basic Operators

The *Kronecker Product* $\boxtimes$ of two matrices $\boldsymbol{A} \in \mathbb{R}^{I \times L}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times M}$ results in a third matrix of size $IJ \times LM$ as follows:

$$\boldsymbol{A} \boxtimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & \ldots & a_{1L}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{I1}\boldsymbol{B} & \ldots & a_{IL}\boldsymbol{B} \end{bmatrix}$$

Now, suppose that we have two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, each partitioned into R sub-matrices such that $\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_R]$ and $\boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_R]$. The *Khatri-Rao Product* $\odot$ of $\boldsymbol{A}$ and $\boldsymbol{B}$

is their partition-wise Kronecker Product:

$$\boldsymbol{A} \odot \boldsymbol{B} = \begin{bmatrix} \boldsymbol{A}_1 \boxtimes \boldsymbol{B}_1 & \dots & \boldsymbol{A}_R \boxtimes \boldsymbol{B}_R \end{bmatrix}$$

Following suite, the *column-wise Khatri-Rao Product* $\odot_c$ of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, having the same number of columns (or in other words, partitioned column-wise into R columns), is their column-wise Kronecker Product:

$$\boldsymbol{A} \odot_c \boldsymbol{B} = \begin{bmatrix} \boldsymbol{a}_1 \boxtimes \boldsymbol{b}_1 & \dots & \boldsymbol{a}_R \boxtimes \boldsymbol{b}_R \end{bmatrix}$$

The *contraction operator* $\bullet_d$, defined for a certain mode $d \in \{1, \dots, N\}$, represents the product between a tensor and a matrix along that mode. For example, suppose that we have $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{L \times M \times N}$, $\boldsymbol{A} \in \mathbb{R}^{I \times L}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times M}$, the first- and second-mode contraction of $\boldsymbol{\mathcal{G}}$ by $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively results in a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times N}$ as follows:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{G}} \underset{1}{\bullet} \boldsymbol{A} \underset{2}{\bullet} \boldsymbol{B} \rightarrow t_{ijn} = \sum_{l=1}^{L} \sum_{m=1}^{M} G_{lmn}\, a_{il} b_{jm} \tag{2.2}$$

The *mode-unfolding* (mode-matricization) of a tensor is reshaping it into a matrix by fixing the index of one of the modes and changing the others. We denote by $\boldsymbol{T}_{(d)}$ the d-th mode-unfolding of tensor $\boldsymbol{\mathcal{T}}$. For instance, for a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$, $\boldsymbol{T}_{(1)} \in \mathbb{R}^{I \times JK}$ represents the first mode-unfolding of $\boldsymbol{\mathcal{T}}$, where the first mode spans the rows of $\boldsymbol{T}_{(1)}$. $\boldsymbol{T}_{(2)} \in \mathbb{R}^{J \times KI}$ and $\boldsymbol{T}_{(3)} \in \mathbb{R}^{K \times JI}$ follow suite as the second and third mode-unfoldings respectively. Sometimes, the targeted mode d of the unfolding is preferred to span the columns of the unfolded matrix $\boldsymbol{T}_{(d)}$, not the rows, which would be the transpose of the aforementioned notations.

2.3 Matrix Factorization

In the following, the data matrix is denoted by $\boldsymbol{M} \in \mathbb{R}^{I \times J}$ $(I \geq J)$, whose two modes represent a subspace each. In general, matrix factorization comes in a lot of different forms, which depends on the desired application [25], but we focus here on two very popular ones, *Singular Value Decomposition* and *Nonnegative Matrix Factorization* [25]. The core of both techniques is that $\boldsymbol{M}$ is decomposed into two main factor matrices, each directly describing one of the subspaces of $\boldsymbol{M}$ and usually having the same number of columns, say $\boldsymbol{A} \in \mathbb{R}^{I \times R}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times R}$, such that $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}$. Sometimes, a diagonal (core) matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{R \times R}$ absorbs the normalization factor of the columns of $\boldsymbol{A}$ and $\boldsymbol{B}$, such that $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{B}^{\mathsf{T}}$. Regarding the other forms of matrix factorization, we note that they are rather constrained or special cases of the form $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}$ that are application-driven, which are out of the scope of this manuscript. However, since nonnegativity is a major constraint in image analysis, and since there could be potential in these forms, we refer to Chapter 1 of the book [25] for a detailed overview.

As such, $\boldsymbol{A}$ is associated with the first mode of $\boldsymbol{M}$, while $\boldsymbol{B}$ is associated with the second. For instance, assuming that the rows of $\boldsymbol{M}$ represent pixels (as samples) and the columns represent spectral reflectance (as spectral features), then, physically speaking, $\boldsymbol{A}$ and $\boldsymbol{B}$ contain patterns related to pixels and spectra respectively, but the quality of these patterns depends on the type of the decomposition. Moreover, the columns of $\boldsymbol{A}$ can be reshaped into matrices that can be shown as grayscale images with some highlighted spatial features.

2.3.1 Singular Value Decomposition and Principal Component Analysis

Singular Value Decomposition (SVD) and Principal Component Analysis (PCA) share the same general expression with one slight difference. Given a matrix $\boldsymbol{M} \in \mathbb{R}^{I \times J}$, SVD decomposes $\boldsymbol{M}$ as it is, while PCA centers the feature space at 0 (that is, the columns of $\boldsymbol{M}$ are subtracted by their means) before applying SVD. In general, SVD is useful for denoising and lossless data compression, and PCA is additionally useful for dimensionality reduction of the feature space. In the following, we suppose that the feature space of $\boldsymbol{M}$ is centered.

SVD uniquely[1] decomposes any matrix $\boldsymbol{M} \in \mathbb{R}^{I \times J}$ (where $I > J$) as follows:

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}} \tag{2.3}$$

$$m_{ij} = \sum_{r=1}^{R} u_{ir}\sigma_{rr}v_{jr} \tag{2.4}$$

where $\boldsymbol{U} \in \mathbb{R}^{I \times I}$ and $\boldsymbol{V} \in \mathbb{R}^{J \times R}$ are orthogonal matrices whose columns represent the left and right singular vectors respectively, $\boldsymbol{\Sigma} \in \mathbb{R}^{I \times R}$ is a diagonal matrix whose diagonal entries are called the singular values and are usually sorted in decreasing order, and generally speaking, we have $R \leq \min(I, J)$. In most of the cases, since $\boldsymbol{\Sigma}$ is diagonal and has extra zero-rows, it is desirable to truncate the corresponding extra columns in $\boldsymbol{U}$ such that we have $\boldsymbol{U} \in \mathbb{R}^{I \times R}$ and $\boldsymbol{\Sigma} \in \mathbb{R}^{R \times R}$, as illustrated in Figure 2.1 such that $I > J$ and $R = J$ (i.e., $\boldsymbol{M}$ is full-rank).

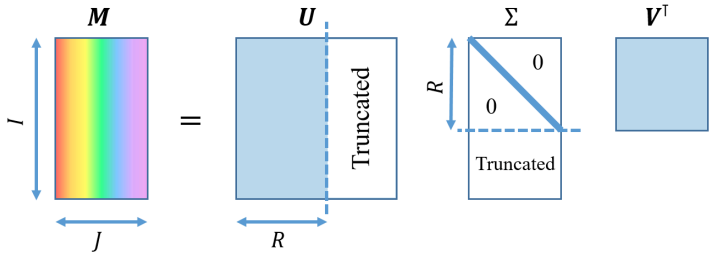


Figure 2.1: Truncated SVD. Since $\boldsymbol{\Sigma}$ is diagonal, then its zero-rows and the corresponding columns of $\boldsymbol{U}$ are truncated. Afterwards, we have $\boldsymbol{U} \in \mathbb{R}^{I \times R}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{R \times R}$, and $\boldsymbol{V} \in \mathbb{R}^{J \times R}$.

SVD is important for many reasons:

- It is *unique* and exact (it can be computed analytically), and applies to any matrix.

[1] Thanks to the orthogonality imposed on matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, the SVD is unique if singular values σ_{rr} are distinct.

- Since $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal, then each of their sets of columns represents the orthonormal basis of the first and second modes of $\boldsymbol{M}$ respectively, then, the columns of $\boldsymbol{U\Sigma}$ are called the principal components (PC). For instance, if $\boldsymbol{M}$ is a matrix of pixels and spectra, then $\boldsymbol{U}$ and $\boldsymbol{V}$ represent the optimal orthonormal subspaces of the pixels and spectral features respectively. $\boldsymbol{U\Sigma}$ then shows the pixel row-vectors as re-expressed (that is, the pixels' new coordinates) in the new spectral feature subspace.

- The singular values of $\boldsymbol{\Sigma}$ represent the statistical weight or distribution (standard deviation) of the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ in decreasing order. The first couple of columns are the most weighted, meaning that the features are more distributed along these directions. As such, the most important part of the data lies there. Conversely, the last columns may correspond to negligible or zero weights, which corresponds to either noisy distributions along their directions or rank-deficiency, and thus can be removed for denoising or dimensionality reduction of the feature space:

  - Figures 2.2 and 2.3 show an illustration and an example of a grayscale image matrix being reconstructed using lower-rank representations (that is, with less singular values and vectors), which is done by truncating the SVD matrices at $R' < R$ components. Compared to the original image (Figure 2.3a), as R' increases (Figures 2.3b-2.3e), we see that the reconstruction shows more details. One could say that $R' = 80$ is enough to visually maintain the details of the scene compared to the full rank that is $R = 130$.

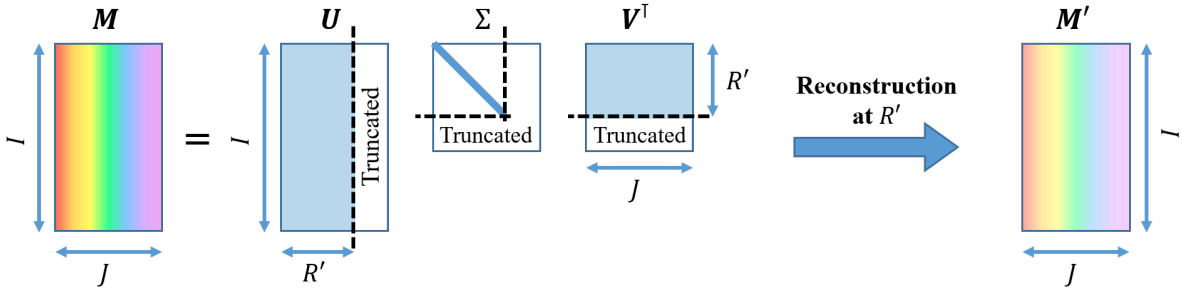
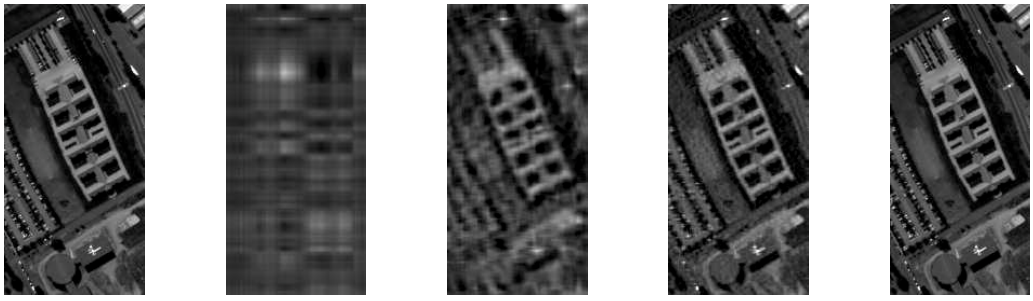


Figure 2.2: Reconstructing a matrix using lower-rank representations



(a) Original (b) $R' = 2$ (c) $R' = 10$ (d) $R' = 30$ (e) $R' = 80$

Figure 2.3: Reconstructing a 270×130 grayscale image matrix (taken from the HSI of Pavia University) using lower-rank representations

– Figure 2.4 shows an illustration of dimensionality reduction using PCA of the HSI of Pavia University, where the spectral feature space dimensionality is reduced from 103 to 4 based on the plot of the singular values of $\boldsymbol{\Sigma}$. As such, the first four PCs (columns) of $\boldsymbol{U\Sigma}$ are reshaped and shown in the figure.

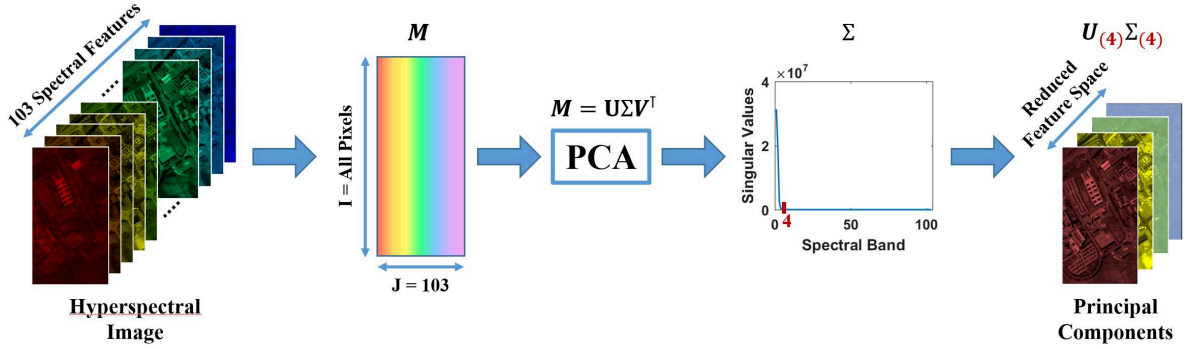


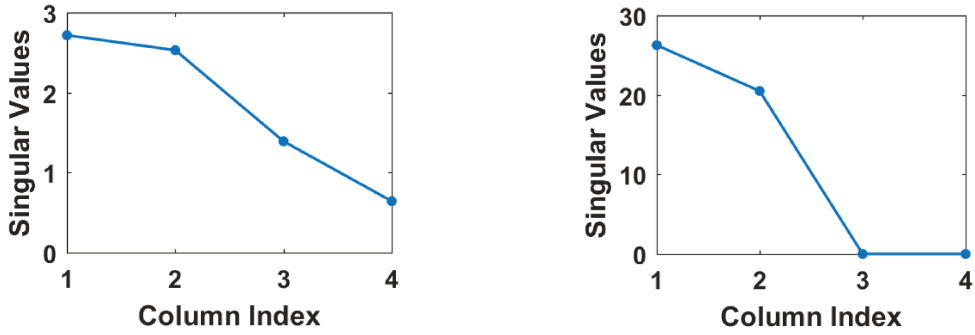
Figure 2.4: Illustration of dimensionality reduction using PCA of the HSI of Pavia University. The plot of $\boldsymbol{\Sigma}$ shows that SVD can be truncated at 4 principal components (PC). $\boldsymbol{U}_{(4)} \in \mathbb{R}^{I\times 4}$ and $\boldsymbol{\Sigma}_{(4)} \in \mathbb{R}^{4\times 4}$ denote the truncated versions of $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$ at 4 PC.

- SVD reveals the *rank* of the matrix, or its intrinsic feature space dimensionality, which is theoretically the number of non-zero singular values in $\boldsymbol{\Sigma}$ and practically the number of those that do not correspond to noise. It is also theoretically defined as the least number of terms, R, for expression (2.4) to be exact. For matrices, we have $R \leq \min(I, J)$.

Figure 2.5 shows the plots of the singular values of two 6×4 synthetic matrices, one created using Matlab's `randn` function (plot (a)), while the other created such that the columns are linearly dependent on two vectors (plot (b)). We can see that the singular values of plot (a) are all non-zero and arranged in decreasing order, which means that the corresponding matrix is full-rank (i.e., $R = 4$), while those of plot (b) have two zero singular values, which means that the corresponding matrix has rank $R = 2$. We note that SVD allows negative entries in its factors, which is not useful when the modeling of the problem requires nonnegativity of the entries (such as the case of NMF in Section 2.3.2).

The elements of SVD can be observed in the diagonalizations of $\boldsymbol{M}\boldsymbol{M}^\mathsf{T}$ and $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$. For instance, the left singular vectors, i.e., $\boldsymbol{U}$, are the eigenvectors of $\boldsymbol{M}\boldsymbol{M}^\mathsf{T} = \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^\mathsf{T}$, and the right singular vectors, i.e., $\boldsymbol{V}$, are the eigenvectors of $\boldsymbol{M}^\mathsf{T}\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\mathsf{T}$. Consequently, the singular values, i.e., $\boldsymbol{\Sigma}$, are the square roots of the eigenvalues of $\boldsymbol{M}\boldsymbol{M}^\mathsf{T}$ and $\boldsymbol{M}^\mathsf{T}\boldsymbol{M}$. In practice, SVD is computed numerically at the base of different algorithms to improve the precision and complexity of the problem.

Finally, SVD has two major properties, *orthogonality* of the factor matrices and *diagonality* of the core matrix, which can not generally coexist in tensor decomposition. Orthogonality was extended to High-Order SVD [36] which we talk about in Section 2.4.1, and diagonality was extended to Canonical Polyadic (CP) decomposition [29] which we talk about in Section 2.4.2.



(a) Linearly independent columns (b) Linearly dependent columns

Figure 2.5: Plots of the singular values of two 6×4 synthetic matrices: (a) The matrix is full-rank $R = 4$, and (b) The matrix has rank $R = 2$.

2.3.2 Nonnegative Matrix Factorization

For many applications, nonnegativity of the entries plays an important role in modeling the problem and physically interpreting the results. Nonnegative Matrix Factorization (NMF) [25] aims at finding the two nonnegative matrices $\boldsymbol{A} \in \mathbb{R}^{I \times R}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times R}$ such that:

$$\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}} \text{ s.t. } \boldsymbol{A} \succeq 0, \boldsymbol{B} \succeq 0 \tag{2.5}$$

where the operator $\succeq$ applies element-wise. In general, NMF is very useful for blind source separation (BSS) of positive real data. For instance, if $\boldsymbol{M}$ is a matrix of pixels and spectra, directly applying NMF can be seen as a BSS technique where the columns of $\boldsymbol{B}$ represent spectral signatures, and those of $\boldsymbol{A}$ represent maps with spatial features.

Unlike SVD, NMF is generally *not unique* (see Section 2.3.2.1 for NMF uniqueness) and can only be approximated, which is usually done by minimizing the following cost function:

$$\begin{aligned} &\underset{\boldsymbol{A},\boldsymbol{B}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{M} - \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}\|_F^2 \\ &\text{s.t. } \boldsymbol{A} \succeq 0, \boldsymbol{B} \succeq 0 \end{aligned} \tag{2.6}$$

where the number of columns of $\boldsymbol{A}$ and $\boldsymbol{B}$ is usually provided as an input. When NMF holds exact, the minimum number of columns is sometimes denoted by R^+ and called the *nonnegative rank*. The cost function (2.6) is not convex, so it is usually solved by breaking it down into convex sub-problems and solving them in an alternating fashion. This technique defines the basic functionality of the algorithm Alternating Least Squares (ALS), which is described and expanded on in details in Section 2.4.5.

Also, unlike $\boldsymbol{U}$ and $\boldsymbol{V}$ from SVD, $\boldsymbol{A}$ and $\boldsymbol{B}$ are generally not orthogonal and do not hold any specific properties beside nonnegativity, so usually additional constraints are imposed on expression (2.6) in order to fit with the desired model or application. Imposing constraints includes those of sparsity, orthogonality, and abundance sum-to-one, and enhances the search for the optimal solution. Finally, extending NMF to tensors may include any form of tensor decomposition where nonnegativity is imposed.

2.3.2.1 Solution Uniqueness

The identifiability of the extracted components, or in other words, the uniqueness of the factor matrices, whether in matrix or tensor decomposition techniques, is a crucial point for any application. To begin with, we start by describing the notion of *essential uniqueness*. If a matrix is unique up to scaling and permutation of its columns, then it is said to be *essentially unique*. For example, for a data matrix $\boldsymbol{M}$ of rank 2, one can write:

$$\boldsymbol{M} = \boldsymbol{A}_{1|2}\boldsymbol{B}_{1|2}^{\mathsf{T}} = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 = \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 + \boldsymbol{a}_1 \otimes \boldsymbol{b}_1 = \boldsymbol{A}_{2|1}\boldsymbol{B}_{2|1}^{\mathsf{T}} \tag{2.7}$$

where $\boldsymbol{A}_{i|j} = [\boldsymbol{a}_i, \boldsymbol{a}_j]$ and $\boldsymbol{B}_{i|j} = [\boldsymbol{b}_i, \boldsymbol{b}_j]$ (which also applies to tensors but for higher orders). Here, if $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, $\boldsymbol{b}_1$, and $\boldsymbol{b}_2$ are identifiable, then $\boldsymbol{A}$ and $\boldsymbol{B}$ are unique up to a permutation. Now, we look at an individual outer product term. For any real nonzero α, we have:

$$\boldsymbol{a} \otimes \boldsymbol{b} = \alpha\alpha^{-1}\boldsymbol{a} \otimes \boldsymbol{b} = (\alpha\boldsymbol{a}) \otimes (\alpha^{-1}\boldsymbol{b}) = \boldsymbol{a}_{\text{Scaled}} \otimes \boldsymbol{b}_{\text{Scaled}} \tag{2.8}$$

Here, if $\boldsymbol{a}$ and $\boldsymbol{b}$ are identifiable, then $\boldsymbol{A}$ and $\boldsymbol{B}$ are unique up to a scaling of their columns. These ambiguities may be reduced by normalizing the columns of the factor matrices and reordering them in decreasing order of their norms.

As for NMF, essential uniqueness is not guaranteed because it suffers from a *rotational ambiguity* of its factor matrices, which is described as follows. Referring to expression (2.5) which is subjected to $\boldsymbol{A}$ and $\boldsymbol{B}$ being nonnegative, it is enough to assume any nonsingular square matrix $\boldsymbol{X}$, and $\boldsymbol{M} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}^{\mathsf{T}}$, such that we have $\tilde{\boldsymbol{A}} = \boldsymbol{A}\boldsymbol{X}$ and $\tilde{\boldsymbol{B}}^{\mathsf{T}} = \boldsymbol{X}^{-1}\boldsymbol{B}^{\mathsf{T}}$, where $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ are both nonnegative. However, if $\boldsymbol{X}$ is guaranteed to be reduced to a generalized permutation matrix, which is a matrix that is formed of only one non-zero element in each column and each row, the rotational ambiguity disappears since $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ become only column-wise permuted versions of $\boldsymbol{A}$ and $\boldsymbol{B}$ repsectively, and NMF is then *essentially unique*.

Usually, imposing constraints on the factor matrices can intrinsically have this consequence on $\boldsymbol{X}$ being reduced to a generalized permutation matrix since imposing constraints (including nonnegativity) helps by restricting the solutions to a few possibilities. In [78], it is shown that the inverse ($\boldsymbol{X}^{-1}$) of a square nonnegative matrix ($\boldsymbol{X}$) is nonnegative if and only if $\boldsymbol{X}$ is a generalized permutation matrix, which consequently means that $\boldsymbol{X}^{-1}$ is also a generalized permutation matrix and $\boldsymbol{X}^{-1} = \boldsymbol{D}\boldsymbol{X}^{\mathsf{T}}$ for some diagonal matrix $\boldsymbol{D}$. Now, if $\boldsymbol{A}$ and $\boldsymbol{B}$, which are supposedly nonnegative, are imposed to be *sufficiently sparse* (which would intrinsically be the case for $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$), then only a generalized permutation matrix $\boldsymbol{X}$ can satisfy the nonnegative constraints in $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$, and NMF is then essentially unique [25].

Finally, on the topic of nonnegativity and sparsity constraints, and the uniqueness of an under-determined linear system of equations (whose components are required to be nonnegative), it is worth mentioning that [16] shows that if there exists a sufficiently sparse solution for the problem, then this solution is necessarily unique. This point is particularly important for our application in Chapter 5.

2.4 Tensor Decomposition

With tensors [29], processing the data is no more restricted to only two diversities or subspaces, or for instance to only sample-feature relationships (e.g., pixel-spectra). Here, it is possible to jointly process all the diversities and subspaces without altering the data array or having to deal with each diversity independently, and this is relevant to many current applications and holds an important future potential where images are being acquired as big data and in multi-modal fashions, some of which we talk about in Section 2.5. This includes data-compression, denoising, BSS, and flexible modeling of the diversities. As such, tensor decomposition is necessary one of the most intuitive and important tools to process multi-modal data.

In the following, the data tensor is denoted by $\boldsymbol{\mathcal{T}}$. For a tensor of order N and dimensions $I_1 \times \cdots \times I_N$, the factors of the decomposition can be indexed by the number of the tensor modes, e.g., $\boldsymbol{H}_d \ \forall\, d \in \{1, \ldots, N\}$, or by alphabetic order, e.g., $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots\}$. In what is related to image analysis thus far, we divide the tensor decomposition techniques into two categories:

- Those that are decomposed into *one* core tensor and N factor matrices:

  This category appears in different forms, but *Tucker Decomposition* is the most general, such that a tensor breaks down into as many factor matrices as the number of its modes, and a core tensor, allowing to uncover underlying information. First, each factor matrix corresponds to one of the tensor modes and holds certain patterns depending on the model and application. Second, the core tensor plays a very important role in *governing the interactions* (i.e., the product) between the columns of the factor matrices.

  There are two main special cases of Tucker Decomposition, (a) *High-Order SVD*, where the factor matrices are orthogonal, and (b) *CP decomposition*, where the core tensor is diagonal. Besides, we have Block Term Decomposition (BTD), where the core tensor is a diagonal of block tensors. We describe each of these techniques in this section in details. Figure 2.6 shows an illustration of the elements of a third-order Tucker Decomposition using the contraction operator (refer to expression (2.9)).

- Those that are decomposed into a *sequence* of core tensors:

  This category contains *Tensor Train (TT) decomposition* [76] and *Tensor Ring (TR) decomposition* [114], which have been respectively used in image analysis for image recognition [13] and Hyperspectral Image Compressive Sensing Reconstruction (HSI-CSR) [18]. Since this category is only briefly described in this manuscript, we refer to the aforementioned references for further details.

  Briefly speaking, $\boldsymbol{\mathcal{T}}$ is decomposed into N third-order core tensors, say $\{\boldsymbol{\mathcal{G}}_1, \ldots, \boldsymbol{\mathcal{G}}_N\}$, where $\boldsymbol{\mathcal{G}}_d \in \mathbb{R}^{R_d \times I_d \times R_{d+1}}$ corresponds to the d-th mode of $\boldsymbol{\mathcal{T}}$ $\forall\, d \in \{1, \ldots, N\}$ (for TR decomposition, we have $R_{N+1} = R_1$), and the set of $\{R_1, \ldots, R_{N+1}\}$ are called the TT-or TR-ranks. Hence, each core tensor is itself a factor, and there are no factor matrices.

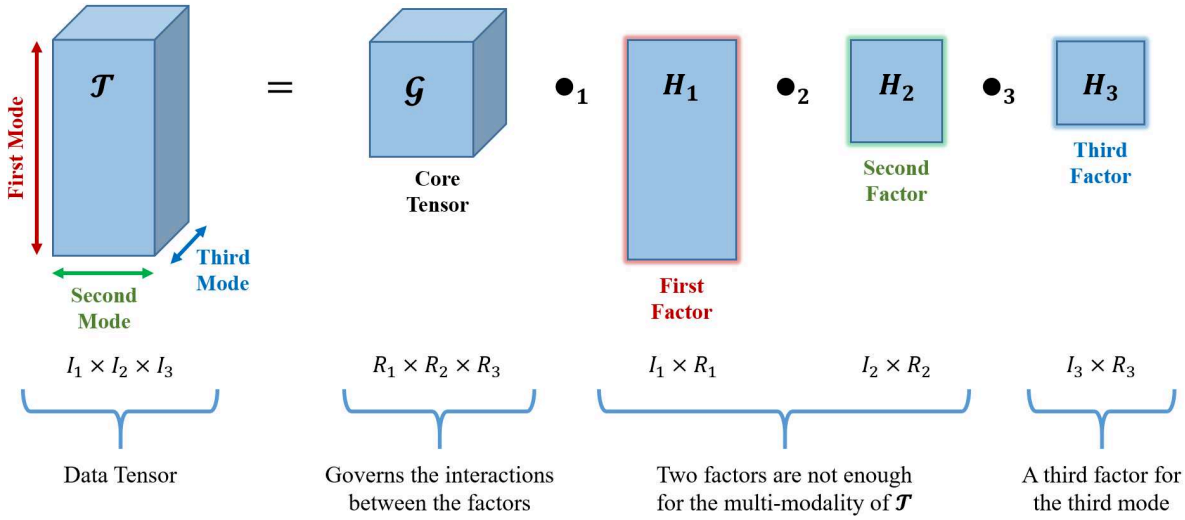


Figure 2.6: Illustration of the elements of a third-order Tucker Decomposition.

2.4.1 Tucker Decomposition and High-Order SVD

The Tucker Decomposition of an Nth-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$, decomposed into a generally full core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times \ldots \times R_N}$ and factor matrices $\{\boldsymbol{H}_1 \in \mathbb{R}^{I_1 \times R_1}, \ldots, \boldsymbol{H}_N \in \mathbb{R}^{I_N \times R_N}\}$, can be expressed using the contraction operator (2.9), which is also expressed element-wise (2.10), as follows:

$$\mathcal{T} = \mathcal{G} \underset{1}{\bullet} \boldsymbol{H}_1 \underset{2}{\bullet} \boldsymbol{H}_2 \ldots \underset{N}{\bullet} \boldsymbol{H}_N \tag{2.9}$$

$$t_{i_1 \ldots i_N} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_N=1}^{R_N} \left(g_{r_1 \ldots r_N} \, h^{(1)}_{i_1 r_1} \, \ldots \, h^{(N)}_{i_N r_N} \right) \tag{2.10}$$

where $\boldsymbol{h}^{(d)}$ represents a column of $\boldsymbol{H}_d$, $\forall\, d \in \{1, \ldots, N\}$. In the following, we switch the tensor order to $N = 3$, which follows analogously for higher orders. In many cases, Tucker Decomposition can be computed by minimizing the error between the tensor and its approximation (usually with additional constraints to fit a model):

$$\underset{\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3, \mathcal{G}}{\operatorname{argmin}} \frac{1}{2} \| \mathcal{T} - \mathcal{G} \underset{1}{\bullet} \boldsymbol{H}_1 \underset{2}{\bullet} \boldsymbol{H}_2 \underset{3}{\bullet} \boldsymbol{H}_3 \|_F^2 \tag{2.11}$$

In general, this decomposition is not identifiable (i.e. essentially unique), so problem (2.11) is usually accompanied with constraints and particular modeling of the factors. In many cases, "Tucker Decomposition" automatically implies that *orthogonality* is imposed on the factor matrices since it is an important property to represent the subspaces of a tensor. For that, High-Order SVD (HOSVD) [36] is one way to guarantee orthogonality, where each factor matrix is the result of applying SVD on the corresponding mode-unfolding of the tensor as follows. Let's denote the d-mode unfolding (matricization) of the tensor by $\boldsymbol{T}_{(d)}$, created when the tensor is reshaped into a matrix of dimensions $(\prod_{k=1|k\neq d}^{3} I_k) \times I_d$. The truncated

SVD of each unfolding is computed such that:

$$\begin{aligned} \boldsymbol{T}_{(d)} &= \boldsymbol{U}_d \boldsymbol{\Sigma}_d \boldsymbol{H}_d^{\mathsf{T}} \\ \forall d &= \{1, 2, 3\} \end{aligned} \tag{2.12}$$

where $\boldsymbol{H}_d \in \mathbb{R}^{I_d \times R_d}$ designates the right singular matrix in the SVD, which is orthogonal. This decomposition can reveal the ranks of the unfoldings, $\{R_1, R_2, R_3\}$, a set which is also referred to as the *multi-linear rank*. The set of right singular matrices are then taken as the new bases, upon which $\boldsymbol{\mathcal{T}}$ is projected to give the core tensor $\boldsymbol{\mathcal{G}}$:

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{T}} \underset{1}{\bullet} \boldsymbol{H}_1^{\mathsf{T}} \underset{2}{\bullet} \boldsymbol{H}_2^{\mathsf{T}} \underset{3}{\bullet} \boldsymbol{H}_3^{\mathsf{T}} \tag{2.13}$$

Since this is SVD, we must have $R_d \leq I_d \; \forall \, d$, so $\boldsymbol{\mathcal{G}}$ is a projected or a compressed version of $\boldsymbol{\mathcal{T}}$. It is worth noting that each scalar of $\boldsymbol{\mathcal{G}}$ can be seen as the weight of interaction between the vectors of the orthonormal bases of $\{\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3\}$ for reconstructing the data tensor $\boldsymbol{\mathcal{T}}$. Hence HOSVD can be expressed as in (2.9), knowing that the factor matrices are orthogonal.

HOSVD is useful for dimensionality reduction of the modes (when $R_d < I_d$), similarly to the case of SVD, by taking out the columns of $\boldsymbol{H}_d$ that correspond to the lowest singular values. Figure 2.7 shows an illustration of third-order HOSVD used for dimensionality reduction of the second and third modes, where $\boldsymbol{I}_{(I_R)}$ represents the identity matrix with dimensions $I_R \times I_R$, and $\boldsymbol{H}_2$ and $\boldsymbol{H}_3$ represent the truncated versions of $\boldsymbol{V}$ and $\boldsymbol{W}$ at R_2 and R_3 PCs respectively. HOSVD is also very useful for lossless compression of huge tensors as processing them can be very demanding [27]. $\boldsymbol{\mathcal{G}}$ is then seen as the compressed tensor and can be processed without loss of information. This case is especially useful for *large images* where the number of pixels can be huge compared to the other modes, for instance, $I_1 \ggg I_2 I_3$. In such cases, the rank of $\boldsymbol{T}_{(1)}$ is less than or equal to $I_2 I_3$ (the minimum of its dimensions), i.e., $R_1 \leq I_2 I_3 \leq I_1$.

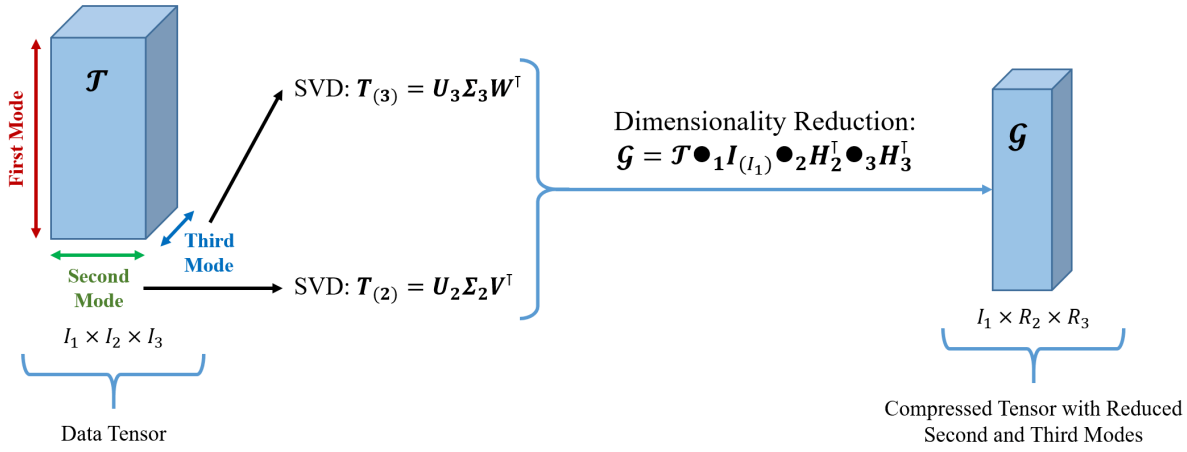


Figure 2.7: Illustration of HOSVD used for the dimensionality reduction of the second and third modes of a third-order tensor. $R_2 < I_2$ and $R_3 < I_3$.

2.4.2 Canonical Polyadic Decomposition

CP decomposition [29] is characterized by having a *diagonal* core tensor, which we denote here by $\mathbf{\Lambda} \in \mathbb{R}^{R \times \ldots \times R}$. Assuming that the scalars of a core tensor represent the weight of interaction between the columns of the factor matrices, then the diagonality of $\mathbf{\Lambda}$ implies that the columns having different indices do not interact. Thanks to this property, CP decomposition is highly used in the literature for BSS problems and feature extraction. Moreover, it can guarantee uniqueness, which we discuss in Section 2.4.3.

An Nth-order *decomposable tensor* $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is a tensor that can be written as the outer product of N vectors $\boldsymbol{h}^{(d)} \in \mathbb{R}^{I_d}$, $\forall\, d \in \{1, \ldots, N\}$:

$$\boldsymbol{\mathcal{D}} \stackrel{\text{def}}{=} \boldsymbol{h}^{(1)} \otimes \ldots \otimes \boldsymbol{h}^{(N)} \tag{2.14}$$

Now, we assume any tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1 \times \ldots \times I_N}$ of order N and tensor rank R. The CP decomposition of $\boldsymbol{\mathcal{T}}$ can be written as a weighted sum of decomposable tensors $\boldsymbol{\mathcal{D}}_r$ (supposing that $\boldsymbol{\mathcal{D}}_r$ are normalized by normalizing the $\{\boldsymbol{h}_r^{(1)}, \ldots, \boldsymbol{h}_r^{(N)}\}$), $\forall\, r \in \{1, \ldots, R\}$ such that:

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \lambda_r \, \boldsymbol{\mathcal{D}}_r = \sum_{r=1}^{R} \lambda_r \left(\boldsymbol{h}_r^{(1)} \otimes \ldots \otimes \boldsymbol{h}_r^{(N)} \right), \tag{2.15}$$

where the *tensor rank*, usually denoted by R, is the least number of terms such that the expression of CP decomposition (2.15) holds exact, which implies that a decomposable tensor is rank-1 ($R = 1$). Referring to expression (2.14) and considering the set of decomposable tensors $\{\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_R\}$, the set of vectors $\{\boldsymbol{h}_1^{(d)}, \ldots, \boldsymbol{h}_R^{(d)}\}$ for a fixed mode $d \in \{1, \ldots, N\}$ can be rearranged as the columns of a matrix such that $\boldsymbol{H}_d = \left[\boldsymbol{h}_1^{(d)}, \ldots, \boldsymbol{h}_R^{(d)}\right] \in \mathbb{R}^{I_d \times R}$. Moreover, the weights $\{\lambda_1, \ldots, \lambda_R\}$ can be rearranged in a diagonal tensor $\mathbf{\Lambda} \in \mathbb{R}^{R \times \ldots \times R}$. Following these notations, CP decomposition can be expressed in Tucker form (2.9) where the core tensor is diagonal.

Moreover, thanks to the diagonality of $\mathbf{\Lambda}$, only the set of columns with a fixed index $r \in \{1, \ldots, R\}$ (i.e., $\{\boldsymbol{h}_r^{(1)}, \ldots, \boldsymbol{h}_r^{(N)}\}$) can interact as observed in the outer products of expression (2.15). The columns of the factor matrices are sometimes called "components", and a set $\{\boldsymbol{h}_r^{(1)}, \ldots, \boldsymbol{h}_r^{(N)}\}$, which can be reconstructed as $\boldsymbol{\mathcal{D}}_r$, fully describes an extracted material. This gives CP decomposition the property of separating the data into individual materials whose qualitative patterns, related to some physical diversities, can be easily observed and associated as column vectors. Figure 2.8 shows an illustration of the properties of CP decomposition for a third-order tensor in Tucker form, and Figure 2.9 illustrates expression (2.15) in the form of decomposable terms.

At this stage, it is important to stress that *there is no scaling ambiguity* in the CP decomposition (2.15), contrary to what is sometimes claimed in the literature. Only the *representation* of tensors $\boldsymbol{\mathcal{D}}_r$ (2.14) by N-uplets of vectors is subject to this indeterminacy. In fact, by definition, tensors are precisely *equivalence classes* with respect to scaling [84, 68, 49, 29]: the sets $\{\boldsymbol{h}_r^{(1)}, \ldots, \boldsymbol{h}_r^{(N)}\}$ and $\{\alpha_1 \boldsymbol{h}_r^{(1)}, \ldots, \alpha_N \boldsymbol{h}_r^{(N)}\}$ represent *the same* tensor provided that $\alpha_1 \ldots \alpha_N = 1$.

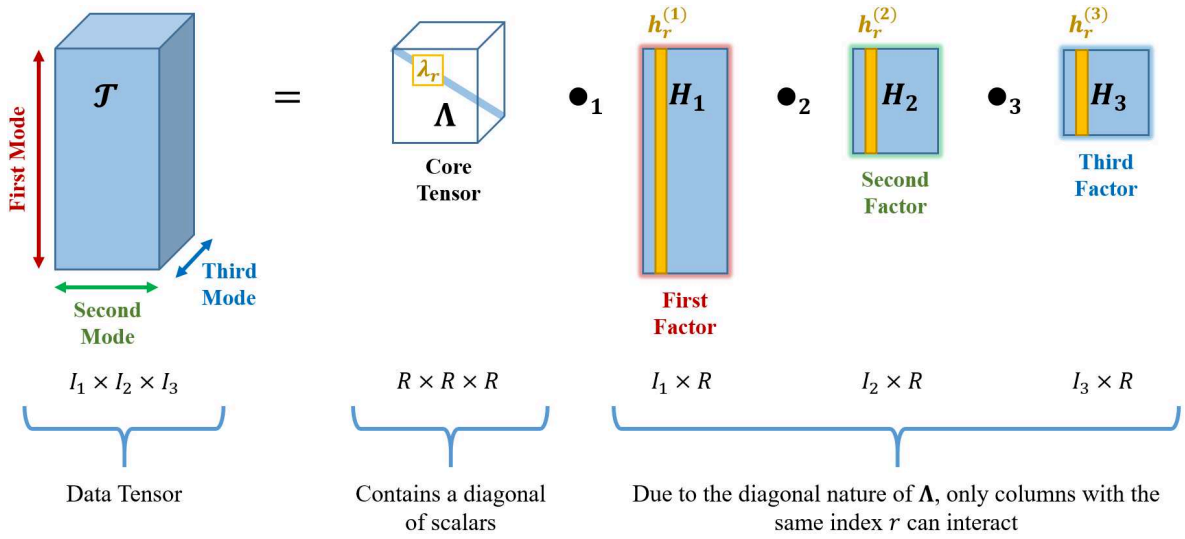


Figure 2.8: Illustration of the elements of a third-order CP decomposition in Tucker form.

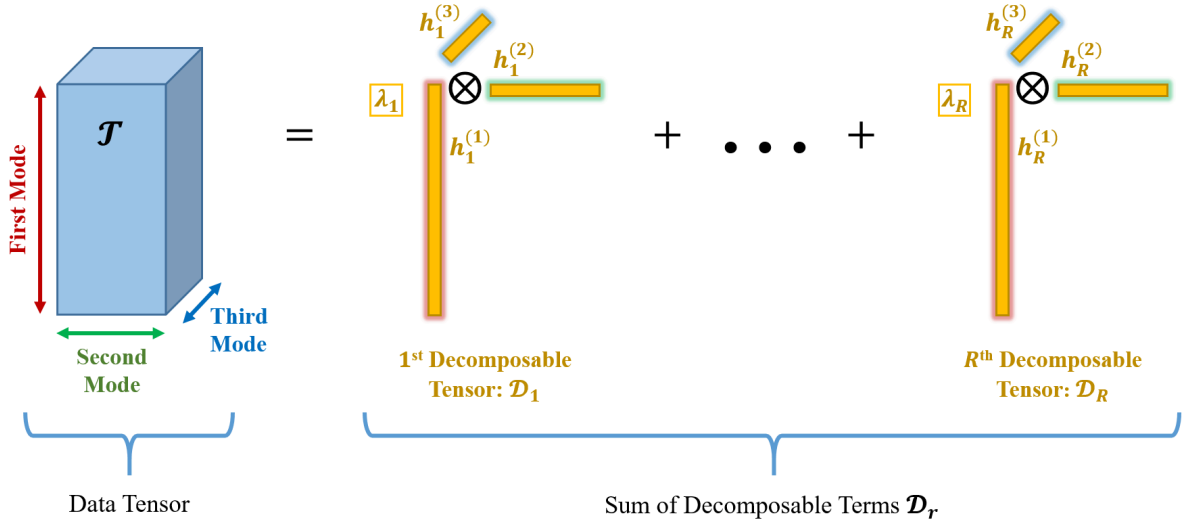


Figure 2.9: Illustration of a third-order CP decomposition as decomposable terms.

2.4.2.1 Tensor Rank

The rank R of the CP decomposition is of particular interest in applications since it is related to the intrinsic subspace dimensionality of multilinear data. It is also related to the property of uniqueness of CP decomposition which is of utmost importance since it eventually allows physical interpretation of relationships among the modes of a tensor. Finding the exact value of the tensor rank is NP-hard [51] and not as direct and analytic as that of the matrix rank, so it is often approximated through trial-and-error and/or with a mix of SVD-based techniques [25]. Contrary to the matrix rank which is upper bounded by the minimum of the matrix's dimensions, it is possible that the tensor rank exceeds the dimensions of the tensor. For example, for an Nth-order tensor where $I_1 \geq \cdots \geq I_N$, a general upper bound of the rank is known to be $R \leq \prod_{d=2}^{N} I_d$ [11].

Furthermore, it is important that the rank is relatively small (with respect to the dimensions of the tensor) for the decomposition to be identifiable. A generic bound on the rank would be to start from the inequality that the number of equations should be greater than the number of unknowns, leading to:

$$R \leq \frac{\prod_{d=1}^{N} I_d}{\sum_{d=1}^{N} I_d - N + 1}, \tag{2.16}$$

which is almost always satisfied since this bound is usually very large, especially when one of the dimensions is the total number of pixels. The identifiability of CP decomposition and the bounds on the rank are discussed in more details in Section 2.4.3.

2.4.2.2 Cost Function

Practically, finding the exact CP decomposition is a hard problem [51], so it is approximated by minimizing the error between the tensor and its approximation. For a third-order tensor, and using the Least Squares (LS) method, the goal is to minimize the cost function:

$$\underset{\boldsymbol{H}_1,\boldsymbol{H}_2,\boldsymbol{H}_3}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{I}}_R \underset{1}{\bullet} \boldsymbol{H}_1 \underset{2}{\bullet} \boldsymbol{H}_2 \underset{3}{\bullet} \boldsymbol{H}_3\|_F^2 \tag{2.17}$$

where $\boldsymbol{\mathcal{I}}_R \in \mathbb{R}^{R\times R\times R}$ is the diagonal tensor of ones, and the values of λ_r are absorbed in the columns of the factor matrices, which in turns are normalized at the end. This cost function is indeed not convex, similar to the case of NMF (2.6), which in fact is a special case of (2.17) for $N = 2$ and where nonnegativity is imposed. As such, problem (2.17) is usually solved by converting it into convex sub-problems in an alternating fashion, each of which is solved for one factor matrix while fixing the others. Alternating Least Squares (ALS) is the most basic algorithm to solve such problems, which we expand on in Section 2.4.5.1. More generally, in Section 2.4.5, we talk about different algorithms for solving CP decomposition.

2.4.2.3 Nonnegativity

When the observation tensor $\boldsymbol{\mathcal{T}}$ contains only real nonnegative entries, it is suitable to impose decomposable tensors $\boldsymbol{\mathcal{D}}_r$ to also be nonnegative. By doing this, we define a *nonnegative rank*, $R^+ \geq R$. This is actually already true for matrices (tensors of order 2). In fact, Herbert E. Robbins exhibited a simple example of a 5×5 matrix having rank 3 but nonnegative rank 4; see [28, 29] for its expression. We refer to nonnegative CP decomposition as NCP decomposition. Following expression (2.17), the cost function for computing NCP decomposition becomes:

$$\begin{aligned} \underset{\boldsymbol{H}_1,\boldsymbol{H}_2,\boldsymbol{H}_3}{\operatorname{argmin}} & \frac{1}{2}\|\boldsymbol{\mathcal{T}} - \boldsymbol{\mathcal{I}}_R \underset{1}{\bullet} \boldsymbol{H}_1 \underset{2}{\bullet} \boldsymbol{H}_2 \underset{3}{\bullet} \boldsymbol{H}_3\|_F^2 \\ & \text{s.t. } \boldsymbol{H}_1 \succeq 0, \boldsymbol{H}_2 \succeq 0, \boldsymbol{H}_3 \succeq 0 \end{aligned} \tag{2.18}$$

Solving this non-convex function follows the concept of ALS, which extends that of NMF (2.6).

There are many applications where nonnegativity is relevant, as to provide better interpretable results when dealing with variables related to physical quantities such as luminance in images, spectra or chemical concentrations [25, 30]. There exist many algorithms aiming at computing the CP decomposition of nonnegative tensors [25, 53]. However, due to measurement noise or modeling errors, the tensor may not be nonnegative or may have a large rank, hence requiring to be approximated. It turns out that, given any real tensor $\boldsymbol{\mathcal{T}}$ of rank R, it is fortunately always possible to find a best nonnegative approximation of $\boldsymbol{\mathcal{T}}$ of given nonnegative rank R^+. This problem is indeed well-posed [69, 81] (which would not be the case in $\mathbb{R}$ instead of $\mathbb{R}^+$). Moreover, imposing nonnegativity helps to limit the space of solutions, improve convergence, and avoid the risk of having degenerate solutions [69, 25].

2.4.3 Solution Uniqueness of CP decomposition

Perhaps the most interesting aspect of CP decomposition is its *uniqueness*, which implies that extracted components and features from multi-modal image data can be uniquely identified. Considering how important this property is, many works have studied the uniqueness of CP decomposition [67, 12, 22, 21, 37, 38, 89, 95, 69, 81, 94, 9] and its approximations [22, 69, 81, 80]. In the following, we talk about these findings and references that can be closely or remotely related to applications in image analysis. This section is divided into three cases that are related to solution uniqueness, unconstrained CP decomposition (Section 2.4.3.1), constrained CP decomposition (the effect of imposing constraints, Section 2.4.3.2), and collinear columns in the factor matrices (Section 2.4.3.3). Furthermore, the latter case leads to some special cases of CP decomposition with particular patterns and derived uniqueness conditions [92, 35, 93, 14], which we talk about afterwards (Section 2.4.4). To begin with, unlike matrix factorization, CP decomposition can be unique under mild conditions and even without constraints.

2.4.3.1 Unconstrained CP decomposition

According to Kruskal [67], an *unconstrained* CP decomposition of a third-order tensor is said to be essentially unique if the following sufficient condition is satisfied:

$$k_1 + k_2 + k_3 \geq 2R + 2 \tag{2.19}$$

where R represents the tensor rank, and k_d represents the *Kruskal rank* of $\boldsymbol{H}_d$. The *Kruskal rank* of a matrix is defined as the maximum number of columns such that any combination of said number of columns is linearly independent. The condition was later extended to Nth-order tensors by Sidiropoulus and Bro [89]:

$$\sum_{d=1}^{N} k_n \geq 2R + N - 1, \tag{2.20}$$

The proof of Kruskal's condition [67] is known to be long and rather inaccessible, but a more accessible proof can be found in [94].

Regarding the difference between the matrix rank ($R_{\cdot}$) and the Kruskal rank ($k_{\cdot}$), we note that assuming a matrix $\boldsymbol{X} \in \mathbb{R}^{I \times J-1}$ ($I > J-1$) whose columns are linearly independent (i.e., $R_{\boldsymbol{X}} = k_{\boldsymbol{X}} = J-1$), if a matrix $\boldsymbol{Y} \in \mathbb{R}^{I \times J}$ is formed by concatenating a column to $\boldsymbol{X}$ such that said column is a linear combination of j columns of $\boldsymbol{X}$, then we have $k_{\boldsymbol{Y}} = j$ following the definition of the Kruskal rank, while $R_{\boldsymbol{Y}}$ can remain as $J-1$. Collinear columns are then a special case of this where $j = 1$. We give an example through the following pseudo-code:

Algorithm 1 Matrix rank and Kruskal rank (Krank) comparative example

Initialize $\boldsymbol{X}$ as a full-rank matrix; *For example:* `X = rand(6,5);`
Create $\boldsymbol{Y}$ by concatenating a column (in $\boldsymbol{X}$) that is a linear combination of the 4th and 5th columns of $\boldsymbol{X}$; *For example:* `Y = cat(2, X, X(:,4) + X(:,5)/2);`
Compute the rank and Kruskal rank of $\boldsymbol{X}$ and $\boldsymbol{Y}$:
return `rank(X)=5; rank(Y)=5; Krank(Y)=2;`

Practical note: One problem with condition (2.20) would be that it depends on parameters related to the factor matrices (i.e., the Kruskal ranks), which are supposedly unknown prior to the decomposition, which can be a problem for practically satisfying the condition. While there is no straight answer to this problem yet, we briefly discuss the following:

- In the least, the condition emphasizes the importance of having a relatively low tensor rank R for the sake of identifiability of the components. After all, Kruskal's sufficient condition shows a much stronger upper bound on the tensor rank than that of (2.16).
- The Kruskal rank is known to be smaller than or equal to the matrix rank ($k_{\boldsymbol{H}_d} \leq R_{\boldsymbol{H}_d}$), which is lower than the minimum of the matrix dimensions, so it is more-or-less possible to make a guess for the Kruskal ranks through the dimensions of the tensor. Here, we note that *a priori* knowledge or modeling of the factor matrices can help. For instance, we know that when the columns of a matrix $\boldsymbol{H}_d$ are linearly independent, we have $k_{\boldsymbol{H}_d} = R_{\boldsymbol{H}_d}$, but when any column is a linear combination of some other columns (including collinearity), we have $k_{\boldsymbol{H}_d} < R_{\boldsymbol{H}_d}$.

2.4.3.2 Constrained CP decomposition

While the aforementioned conditions concern the unconstrained CP decomposition, the conditions may be relaxed or can be different when constraints are imposed [16, 93]. Moreover, tensor decomposition can still adopt the advantages of imposing constraints in the case of matrices. For instance, nonnegativity constraints are important for the physical interpretation in almost any application of image analysis, and sparsity of the factors is physically significant for many applications where the extracted features are assumed to be few per sample (such as the case of spectral unmixing). In general, nonnegativity limits the space of solutions, improves convergence, and removes the risk of having degenerate solutions [69, 25], and the sparseness of the factor matrices can improve the uniqueness of a solution (see Section 2.3.2.1 for the uniqueness of NMF) and can even guarantee its uniqueness in the presence of nonnegative constraints [16, 25].

Speaking of nonnegativity, we recall that imagery data are in the most cases nonnegative. Fortunately, it is now shown that given a real nonnegative tensor $\boldsymbol{\mathcal{T}}$ of nonnegative rank $R^+ > r$ (i.e., for any given value r), finding the best nonnegative rank-r *approximation* $\hat{\boldsymbol{\mathcal{T}}}$ such that:

$$\underset{\hat{\boldsymbol{\mathcal{T}}}}{\operatorname{argmin}} \|\boldsymbol{\mathcal{T}} - \hat{\boldsymbol{\mathcal{T}}}\|_F^2, \tag{2.21}$$

is a well-posed problem. The solution exists, and it is almost always unique [69, 81].

2.4.3.3 Collinearity and Kruskal's Condition

At this stage, it is important to talk about a special case of CP decomposition, where there are collinear columns (or collinear loadings) in the factor matrices. This has a direct effect both on the CP model and its uniqueness and can be observed through Kruskal's condition even though the latter is sufficient and not necessary for uniqueness. For instance, in some applications of multi-modal image analysis (see Chapter 5), it can be expected in theory to obtain two collinear columns in a factor matrix. In the following, we describe the direct consequences of collinear loadings on the uniqueness of CP decomposition. First, we start with the case of third-order tensors, then we talk about higher-order ones.

As it is noted in [94], a necessary condition for the uniqueness of unconstrained CP decomposition of third-order tensors is that the factor matrices should not have collinear columns. First, collinear columns in any of the factor matrices may cause a local rotational ambiguity at the involved columns. Second, collinear columns correspond to a Kruskal rank of 1, which violates Kruskal's condition (2.19) for third-order tensors. These observations can be further demonstrated as follows:

- Assuming that the factor matrices $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are full-rank and two columns of $\boldsymbol{H}_3$ are collinear, then $k_3 = 1$. This reduces expression (2.19) to

  $$k_1 + k_2 \geq 2R + 1,$$

  which is impossible since the Kruskal ranks of the factor matrices are upper-bounded by R (i.e., $k_1 + k_2 \leq 2R$). This means that uniqueness is not guaranteed.

- Suppose that columns s and t ($s \neq t$) of $\boldsymbol{H}_3$ are collinear, then we can write the following [94]:

$$\begin{aligned}
\boldsymbol{\mathcal{D}}_s + \boldsymbol{\mathcal{D}}_t &= \boldsymbol{h}_s^{(1)} \otimes \boldsymbol{h}_s^{(2)} \otimes \boldsymbol{h}_s^{(3)} + \boldsymbol{h}_t^{(1)} \otimes \boldsymbol{h}_t^{(2)} \otimes \boldsymbol{h}_t^{(3)} \\
&= \alpha \boldsymbol{h}_s^{(1)} \otimes \boldsymbol{h}_s^{(2)} \otimes \boldsymbol{h}_t^{(3)} + \boldsymbol{h}_t^{(1)} \otimes \boldsymbol{h}_t^{(2)} \otimes \boldsymbol{h}_t^{(3)} \\
&= (\alpha \boldsymbol{h}_s^{(1)} \otimes \boldsymbol{h}_s^{(2)} + \boldsymbol{h}_t^{(1)} \otimes \boldsymbol{h}_t^{(2)}) \otimes \boldsymbol{h}_t^{(3)} \\
&= (\boldsymbol{H}_{1_{s|t}} \boldsymbol{H}_{2_{s|t}}^{\mathsf{T}}) \otimes \boldsymbol{h}_t^{(3)} \\
&= (\boldsymbol{H}_{1_{s|t}} \boldsymbol{X} \boldsymbol{X}^{-1} \boldsymbol{H}_{2_{s|t}}^{\mathsf{T}}) \otimes \boldsymbol{h}_t^{(3)} \\
&= (\tilde{\boldsymbol{H}}_{1_{s|t}} \tilde{\boldsymbol{H}}_{2_{s|t}}^{\mathsf{T}}) \otimes \boldsymbol{h}_t^{(3)}
\end{aligned} \tag{2.22}$$

for any 2×2 nonsingular matrix $\boldsymbol{X}$, where matrix $\boldsymbol{H}_{1_{s|t}} = \left[\alpha \boldsymbol{h}_s^{(1)}, \boldsymbol{h}_t^{(1)}\right]$ and $\boldsymbol{H}_{2_{s|t}} = \left[\boldsymbol{h}_s^{(2)}, \boldsymbol{h}_t^{(2)}\right]$. Then the CP decomposition is not essentially unique due to the local rotational ambiguity at the columns s and t in the factor matrices.

Here, one can argue that using nonnegativity and sparsity constraints can relax the conditions and reduce the effect of the local rotational ambiguity, similar to the discussion of the uniqueness of NMF.

However, for tensors of order higher than 3, it is interesting that CP decomposition can still be unique even with the existence of collinear columns in the factor matrices. For instance, some conditions for fouth-order tensors were discussed in details in [14]. Here, we only make some simple and intuitive demonstration of this observation, and we refer to [14] for the proofs and details. Tensors of order higher than 4 follow analogously:

- Looking at (2.20), we have:

  $$k_1 + k_2 + k_3 + k_4 \geq 2R + 3.$$

  If $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, and $\boldsymbol{H}_3$ are full and two columns of $\boldsymbol{H}_4$ are collinear, then $k_4 = 1$. This reduces the inequality to

  $$k_1 + k_2 + k_3 \geq 2R + 2,$$

  which interestingly reduces to the case of third-order tensors (2.19).

- Suppose that columns s and t ($s \neq t$) of $\boldsymbol{H}_4$ are collinear, then we can write the following:

  $$\begin{aligned} \boldsymbol{\mathcal{D}}_s + \boldsymbol{\mathcal{D}}_t &= \boldsymbol{h}_s^{(1)} \otimes \boldsymbol{h}_s^{(2)} \otimes \boldsymbol{h}_s^{(3)} \otimes \boldsymbol{d}_s + \boldsymbol{h}_t^{(1)} \otimes \boldsymbol{h}_t^{(2)} \otimes \boldsymbol{h}_t^{(3)} \otimes \boldsymbol{d}_t \\ &= \alpha \boldsymbol{h}_s^{(1)} \otimes \boldsymbol{h}_s^{(2)} \otimes \boldsymbol{h}_s^{(3)} \otimes \boldsymbol{d}_t + \boldsymbol{h}_t^{(1)} \otimes \boldsymbol{h}_t^{(2)} \otimes \boldsymbol{h}_t^{(3)} \otimes \boldsymbol{d}_t \\ &= (\alpha \boldsymbol{h}_s^{(1)} \otimes \boldsymbol{h}_s^{(2)} \otimes \boldsymbol{h}_s^{(3)} + \boldsymbol{h}_t^{(1)} \otimes \boldsymbol{h}_t^{(2)} \otimes \boldsymbol{h}_t^{(3)}) \otimes \boldsymbol{d}_t \end{aligned} \tag{2.23}$$

  which does not simply reduce to a local rotational ambiguity between the s-th and t-th outer products unless, for instance, further collinearity exists inside the triplets in the parentheses. Instead, it reduces to a local third-order CP decomposition, where there is no rotation matrix $\boldsymbol{X}$ whose inverse $\boldsymbol{X}^{-1}$ can counteract the effect of the rotation.

Linear dependency and collinearity between the columns of the factor matrices in CPD is an issue that has been investigated further in the literature due to their relevance for many applications (not necessarily in image analysis) [35, 93, 92], where we find that other techniques were derived such as: (a) Block Term Decomposition (BTD) into $(L_r, L_r, 1)$-terms (also known as $(L_r, L_r, 1)$-decomposition), (b) PARAllel profiles with LINear Dependencies (PARALIND), and (c) CONstrained FACtors (CONFAC). Each of them has its derived uniqueness conditions. In the following, we talk in some details about $(L, L, 1)$-decomposition, but we only briefly go through PARALIND and CONFAC since they are out of the scope of this manuscript in terms of image analysis, but there might be some future potential.

2.4.4 Derived Cases of CP decomposition

In this section, the description is done for third-order tensors denoted by $\boldsymbol{\mathcal{T}}$ of dimensions $I \times J \times K$, where the factor matrices are denoted by $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$ representing the first, second, and third modes respectively.

2.4.4.1 $(L_r, L_r, 1)$-decomposition

$(L_r, L_r, 1)$-terms is a special case of the more general form that is the BTD into (L_r, M_r, N_r) terms. The general form of BTD refers to a tensor decomposition where the core tensor $\boldsymbol{\mathcal{G}}$, in expression (2.9), is formed of R nonzero tensor blocks rearranged along the diagonal (see Figure 2.10), each of dimensions $L_r \times M_r \times N_r \ \forall r \in \{1, \ldots, R\}$. Consequently, the factor matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are partitioned as follows:

$\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ have dimensions $I \times (\sum_{r=1}^{R} L_r)$, $J \times (\sum_{r=1}^{R} M_r)$, and $K \times (\sum_{r=1}^{R} N_r)$ respectively. If $L_r = L$, $M_r = M$, and $N_r = N \ \forall r \in \{1, \ldots, R\}$, then we have $\boldsymbol{A} \in \mathbb{R}^{I \times RL}$, $\boldsymbol{B} \in \mathbb{R}^{J \times RM}$, and $\boldsymbol{C} \in \mathbb{R}^{K \times RN}$. $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are then partitioned into R sub-matrices where each partition is denoted by $\boldsymbol{A}_r \in \mathbb{R}^{I \times L_r}$, $\boldsymbol{B}_r \in \mathbb{R}^{J \times M_r}$, and $\boldsymbol{C}_r \in \mathbb{R}^{K \times N_r}$, and:

$$\boldsymbol{A} = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_R]\,;\ \boldsymbol{B} = [\boldsymbol{B}_1, \ldots, \boldsymbol{B}_R]\,;\ \boldsymbol{C} = [\boldsymbol{C}_1, \ldots, \boldsymbol{C}_R]$$

BTD into (L_r, M_r, N_r) can be illustrated into two forms: (a) as one Tucker expression using one core tensor that contains the blocks (Figure 2.10), and (b) as a sum of Tucker expressions by viewing the block tensors as core tensors (Figure 2.11). In this sense, CP decomposition can be seen as a special case of BTD where BTD simply decomposes into $(1, 1, 1)$-terms (i.e., the nonzero tensor blocks $\boldsymbol{\mathcal{G}}_r$ reduce to scalars).

In particular, we focus here on the case of $(L_r, L_r, 1)$-decomposition for its special representation and since it has a relationship with CP decomposition. In $(L_r, L_r, 1)$-decomposition, each partition pair $(\boldsymbol{A}_r,\ \boldsymbol{B}_r)$ is a pair of sub-matrices having equal number of columns L_r, while the corresponding partition $\boldsymbol{c}_r$ is one column itself. Since $\boldsymbol{C}$ is partitioned into columns (instead of sub-matrices), the block tensors $\boldsymbol{\mathcal{G}}_r$ reduce to matrices of dimensions $L_r \times L_r$. As such, $(L_r, L_r, 1)$-decomposition can be expressed as follows:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{G}} \underset{1}{\bullet} \boldsymbol{A} \underset{2}{\bullet} \boldsymbol{B} \underset{3}{\bullet} \boldsymbol{C} = \sum_{r=1}^{R} \boldsymbol{\mathcal{G}}_r \underset{1}{\bullet} \boldsymbol{A}_r \underset{2}{\bullet} \boldsymbol{B}_r \underset{3}{\bullet} \boldsymbol{c}_r = \sum_{r=1}^{R} (\boldsymbol{A}_r \boldsymbol{G}_r \boldsymbol{B}_r^{\mathsf{T}}) \otimes \boldsymbol{c}_r \tag{2.24}$$

where if $\boldsymbol{G}_r$ is absorbed by either $\boldsymbol{A}_r$ or $\boldsymbol{B}_r$, the expression can be:

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} (\boldsymbol{A}_r \boldsymbol{B}_r^{\mathsf{T}}) \otimes \boldsymbol{c}_r \quad = \sum_{r=1}^{R} \boldsymbol{E}_r \otimes \boldsymbol{c}_r \tag{2.25}$$

where $\boldsymbol{E}_r \in \mathbb{R}^{I \times J}$ has a matrix rank $\leq L_r$. Expression (2.25) is illustrated in Figure 2.12.

At this stage, the correspondence in terms of collinearity between $(L_r, L_r, 1)$-decomposition and CP decomposition can be seen as follows. If we suppose that $\boldsymbol{C}$ is recreated such that each

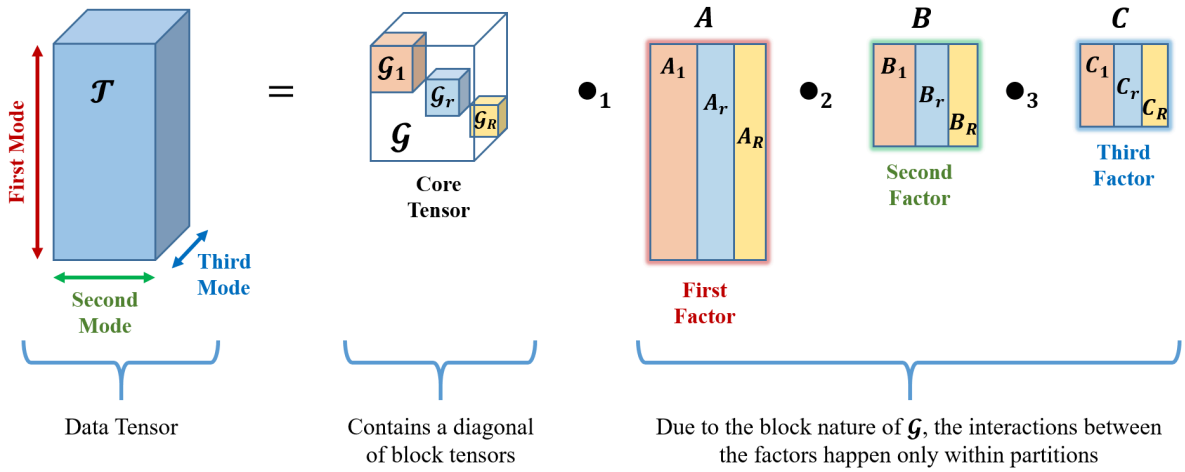


Figure 2.10: Illustration of the elements of a third-order BTD in general Tucker form.

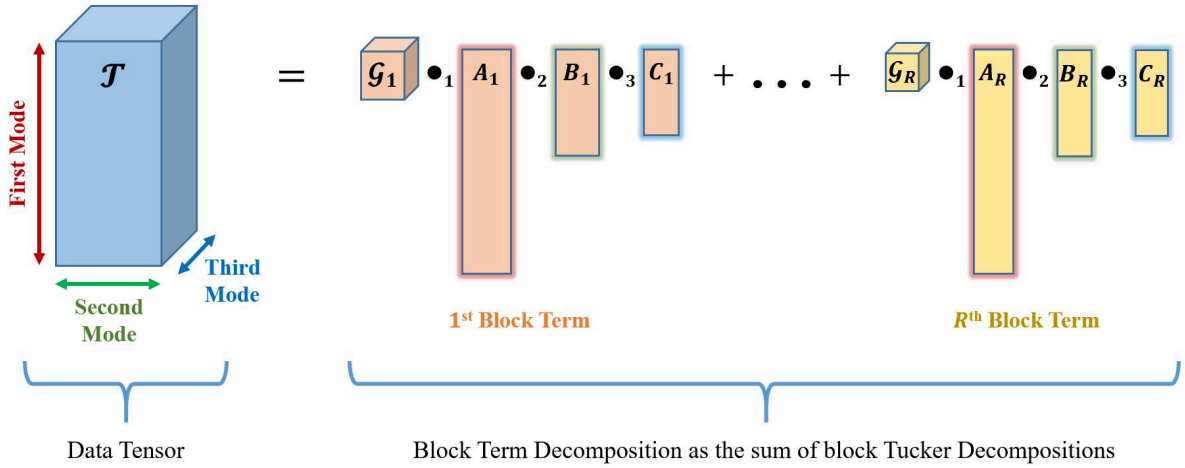


Figure 2.11: Illustration of a third-order BTD as the sum of block Tucker Decompositions.

column vector $\boldsymbol{c}_r$ is repeated L_r times, then $(L_r, L_r, 1)$-decomposition is equivalent to a CP decomposition where the tensor rank is $\sum_{r=1}^{R} L_r$. Going backwards, in a CP decomposition of a third-order tensor, if the columns of indices $\{s, t, u\}$ of factor matrix $\boldsymbol{C}$ are collinear, those of $\boldsymbol{A}$ and $\boldsymbol{B}$ can be grouped into partition sub-matrices $\boldsymbol{A}_{s|t|u}$ and $\boldsymbol{B}_{s|t|u}$ as follows:

$$\begin{aligned}\boldsymbol{\mathcal{D}}_s + \boldsymbol{\mathcal{D}}_t + \boldsymbol{\mathcal{D}}_u &= \boldsymbol{a}_s \otimes \boldsymbol{b}_s \otimes \boldsymbol{c}_s + \boldsymbol{a}_t \otimes \boldsymbol{b}_t \otimes \boldsymbol{c}_t + \boldsymbol{a}_u \otimes \boldsymbol{b}_u \otimes \boldsymbol{c}_u \\ &= \boldsymbol{a}_s \otimes \boldsymbol{b}_s \otimes \boldsymbol{c}_s + \lambda_t \boldsymbol{a}_t \otimes \boldsymbol{b}_t \otimes \boldsymbol{c}_s + \lambda_u \boldsymbol{a}_u \otimes \boldsymbol{b}_u \otimes \boldsymbol{c}_s \\ &= (\boldsymbol{a}_s \otimes \boldsymbol{b}_s + \lambda_t \boldsymbol{a}_t \otimes \boldsymbol{b}_t + \lambda_u \boldsymbol{a}_u \otimes \boldsymbol{b}_u) \otimes \boldsymbol{c}_s \\ &= (\boldsymbol{A}_{s|t|u} \boldsymbol{B}_{s|t|u}^{\mathsf{T}}) \otimes \boldsymbol{c}_s = (\boldsymbol{A}_r \boldsymbol{B}_r^{\mathsf{T}}) \otimes \boldsymbol{c}_r\end{aligned} \tag{2.26}$$

which resembles one summand of (2.25). $\boldsymbol{A}_{s|t|u} = [\boldsymbol{a}_s, \lambda_t \boldsymbol{a}_t, \lambda_u \boldsymbol{a}_u]$ and $\boldsymbol{B}_{s|t|u} = [\boldsymbol{b}_s, \boldsymbol{b}_t, \boldsymbol{b}_u]$.

Regarding computation, $(L_r, L_r, 1)$-decomposition shares the same cost function as that of Tucker (2.11), however its computation is rather similar to CP decomposition in terms of updates, which can also be computed using ALS as described in Section 2.4.5.1.

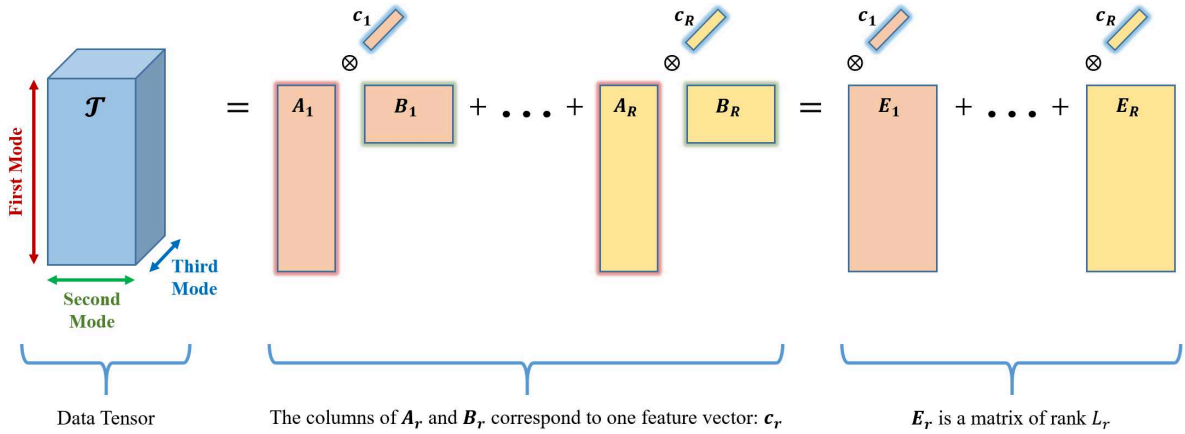


Figure 2.12: Illustration of $(L_r, L_r, 1)$-decomposition.

In terms of physical significance, sometimes it is more convenient to extract a feature vector (e.g., spectral reflectance), say $\boldsymbol{c}_r$, which corresponds to more than one pair of feature columns in each of $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$ (e.g., the two pixel modes), so directly using the CP model risks its identifiability due to collinearity. At this point, it is good to talk about the conditions of essential uniqueness of $(L_r, L_r, 1)$-decomposition.

Several essential uniqueness conditions are discussed in [35], but they all boil down to an extension of condition (2.27) for partitioned matrices. First, the k'-*rank* of a partitioned matrix is defined as the maximum number of *partitions* such that the columns represented by any combination of said number of partitions are linearly independent. Now, in a parallel fashion to (2.19), a sufficient condition for $(L_r, L_r, 1)$ uniqueness is:

$$k'_{\boldsymbol{A}} + k'_{\boldsymbol{B}} + k_{\boldsymbol{C}} \geq 2R + 2 \tag{2.27}$$

A necessary condition however is that $\boldsymbol{A}$ and $\boldsymbol{B}$ should not have collinear columns. For instance, the representation of $(L_r, L_r, 1)$-decomposition as a model that reduces collinearity of CP decomposition already assumes that the columns of $\boldsymbol{C}$ are not collinear, so if $\boldsymbol{C}$ is full rank (the columns are linearly independent) and $K \geq R$, then $k_{\boldsymbol{C}} = R$. Now if two columns of either $\boldsymbol{A}$ or $\boldsymbol{B}$ are collinear, whether within the same partition or not, the corresponding $k' = 1$, which violates condition (2.27). In fact, if the collinear columns, say in $\boldsymbol{A}$, belong to the same partition, the corresponding ones in $\boldsymbol{B}$ can be added into one. Otherwise, uniqueness is not guaranteed, so imposing constraints like nonnegativity or sparsity can help depending on the problem and its conditions.

2.4.4.2 PARALIND and CONFAC

We have seen that linear dependency and collinearity in the factor matrices are directly related to low k-rank and k'-rank values, hence potentially rendering the sufficient conditions for uniqueness deficient. $(L_r, L_r, 1)$-decomposition can be a good alternative to CP decomposition if collinearity is restricted to one matrix, but when more than two matrices are

concerned by *linear dependency* or *collinearity*, $(L_r, L_r, 1)$-decomposition becomes restricting as a model. For that, PARALIND and CONFAC were proposed as (a) *constrained versions* of CP decomposition for linear dependency and collinearity respectively, and (b) more flexible representations of these properties using the so-called *constraint matrices* [93]. In the following, we only briefly describe the two techniques since this manuscript does not deal with them in terms of image analysis, but we believe that they could be a good reference for future works since they are related to special cases of CP decomposition, which is widely used in image analysis. For further details and examples, we refer to [93].

The constraints are represented by the set of *pre-specified* and *fixed* constraint matrices $\{\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Omega}\}$, such that instead of the set $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$, now we have $\{\boldsymbol{A\Psi}, \boldsymbol{B\Phi}, \boldsymbol{C\Omega}\}$, such that:

- $\boldsymbol{A} \in \mathbb{R}^{I \times R_1}$ and $\boldsymbol{\Psi} \in \mathbb{R}^{R_1 \times R}$ $(R_1 \leq R)$
- $\boldsymbol{B} \in \mathbb{R}^{J \times R_2}$ and $\boldsymbol{\Phi} \in \mathbb{R}^{R_2 \times R}$ $(R_2 \leq R)$
- $\boldsymbol{C} \in \mathbb{R}^{K \times R_3}$ and $\boldsymbol{\Omega} \in \mathbb{R}^{R_3 \times R}$ $(R_3 \leq R)$

Briefly speaking, the constraint matrices $\boldsymbol{\Psi}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Omega}$ contain the patterns of linear dependency and collinearity in $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ respectively. In this section, we are more interested in collinearity, so we focus on the case of CONFAC, which is a special case of PARALIND but where the columns of the constraint matrices are restricted to (repeated) unit vectors. For example on $\boldsymbol{A}$ and $\boldsymbol{\Psi}$ without loss of generality:

$$\text{Say: } \boldsymbol{A\Psi} = \begin{bmatrix} 1 & 5 & 1 \\ 2 & 6 & 2 \\ 3 & 7 & 3 \\ 4 & 8 & 4 \end{bmatrix}; \text{ Then: } \boldsymbol{A} = \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix} \text{ and } \boldsymbol{\Psi} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{2.28}$$

Regarding uniqueness conditions, the work of [93] differentiates between two concepts of uniqueness from the perspective of one factor matrix (say $\boldsymbol{A}$):

- Essential uniqueness: $\boldsymbol{A}$ is said to be essentially unique if it is unique up to column-wise scaling and permutation.
- Partial uniqueness: $\boldsymbol{A}$ is said to be partially unique if it is unique up to partition-wise scaling and permutation (that is, the way $\boldsymbol{A}$ is partitioned is unique) where each partition is identified up to its linear span. In this sense, any form of BTD is partially unique for all its factor matrices since, by definition, their partitioning is fixed by (L_r, M_r, N_r).

In general, a lot of necessary and sufficient conditions can be found in [93], most of which are derived using the permutation lemma of Kruskal [67].

Moreover, it is important to note that uniqueness is discussed over the set of factor matrices $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$, and not the constraint matrices, which means that a factor matrix is unique up to

scaling and permutation over its own un-repeated columns while the set $\{\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Omega}\}$ is fixed. This means that, in example (2.28) for instance, if $\tilde{\boldsymbol{A}}$ is a permuted version of $\boldsymbol{A}$ (as in if the columns are permuted), then the product $\tilde{\boldsymbol{A}}\boldsymbol{\Psi}$ is not necessarily equal to $\boldsymbol{A}\boldsymbol{\Psi}$.

Regarding the computation of PARALIND and CONFAC, some approaches using ALS are proposed in the literature, for which we refer to [34, 15]. Finally, we note that if the constraint matrices $\{\boldsymbol{\Psi}, \boldsymbol{\Phi}, \boldsymbol{\Omega}\}$ are all identity matrices, then PARALIND and CONFAC are equivalent to CP decomposition.

2.4.5 Computation of CP decomposition

There are many algorithms in the literature to compute CP decomposition. The most basic and popular one is Alternating Least Squares (ALS), which converts the main cost function into sub-problems solved for each factor matrix alone while considering that the others are fixed. One iteration of ALS is complete when each factor matrix is updated once. These sub-problems are indeed convex, and they are solved in an alternating fashion until convergence or a stopping criterion is reached. ALS has a lot of extensions such as Nonnegative ALS (NALS) for nonnegative constraints [25], Projected-Compressed ALS (ProCo ALS) [27] for huge tensors (compression) and constraints that require hard threshoding such as nonnegativity (projection), and the recently proposed Alternating Optimization - Alternating Direction Method of Multipliers (AO-ADMM) [53] which is efficient and more flexible with constraints. In the following, we describe these algorithms for third-order tensors. Higher-order tensors follow analogously.

2.4.5.1 Alternating Least Squares

In ALS, one iteration is done by solving the following sub-problems:

$$\begin{cases} \boldsymbol{H}_1 = \underset{\boldsymbol{H}_1}{\operatorname{argmin}} \dfrac{1}{2}\|\boldsymbol{T}_{(1)} - \boldsymbol{W}_1 \boldsymbol{H}_1^{\mathsf{T}}\|_F^2 \\ \boldsymbol{H}_2 = \underset{\boldsymbol{H}_2}{\operatorname{argmin}} \dfrac{1}{2}\|\boldsymbol{T}_{(2)} - \boldsymbol{W}_2 \boldsymbol{H}_2^{\mathsf{T}}\|_F^2 \\ \boldsymbol{H}_3 = \underset{\boldsymbol{H}_3}{\operatorname{argmin}} \dfrac{1}{2}\|\boldsymbol{T}_{(3)} - \boldsymbol{W}_3 \boldsymbol{H}_3^{\mathsf{T}}\|_F^2 \end{cases} \tag{2.29}$$

where $\boldsymbol{T}_{(d)}$ represents the d-mode unfolding of $\boldsymbol{\mathcal{T}}$, and $\boldsymbol{W}_d$ contains the column-wise Khatri-Rao product of all the factor matrices excluding $\boldsymbol{H}_d$ (for example, for $d = 1$, $\boldsymbol{T}_{(1)}$ has dimensions $I_2 I_3 \times I_1$, and $\boldsymbol{W}_1 = \boldsymbol{H}_3 \odot_c \boldsymbol{H}_2$ has dimensions $I_2 I_3 \times R$). The factor matrices are then updated as follows until convergence or a stopping criterion is met:

$$\begin{cases} \boldsymbol{H}_1 \leftarrow (\boldsymbol{W}_1{}^{\dagger} \cdot \boldsymbol{T}_{(1)})^{\mathsf{T}} \\ \boldsymbol{H}_2 \leftarrow (\boldsymbol{W}_2{}^{\dagger} \cdot \boldsymbol{T}_{(2)})^{\mathsf{T}} \\ \boldsymbol{H}_3 \leftarrow (\boldsymbol{W}_3{}^{\dagger} \cdot \boldsymbol{T}_{(3)})^{\mathsf{T}} \end{cases} \tag{2.30}$$

where † designates the Moore–Penrose pseudoinverse of a matrix.

In the case of $(L_r, L_r, 1)$-decomposition, the updates are very similar to those of CP decomposition with a major difference in the computation of $\boldsymbol{W}_d$ that accounts to the partitioning of the factor matrices.

$$\begin{cases} \boldsymbol{W}_3 = [(\boldsymbol{B}_1 \odot_c \boldsymbol{A}_1)\mathbf{1}_{L_1}, \ldots, (\boldsymbol{B}_R \odot_c \boldsymbol{A}_R)\mathbf{1}_{L_R}] \\ \boldsymbol{W}_1 = \boldsymbol{C} \odot \boldsymbol{B} \\ \boldsymbol{W}_2 = \boldsymbol{C} \odot \boldsymbol{A} \end{cases} \tag{2.31}$$

where $\mathbf{1}_{L_R}$ represents a vector of ones and dimension L_R. $\boldsymbol{W}_3$ is updated using the column-wise Khatri-Rao product between partitions, while $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are updated using the partition-wise Khatri-Rao product. After that, $\{\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3\}$ can be updated following (2.30). As for the core tensor $\boldsymbol{\mathcal{G}}$, its block tensors $\boldsymbol{G}_r \in \mathbb{R}^{L_r \times L_r \times 1}$ are diagonal matrices whose entries absorb the normalization factors of $\boldsymbol{A}_r$, $\boldsymbol{B}_r$, and $\boldsymbol{c}_r$ after the termination of the algorithm.

2.4.5.2 Nonnegative Alternating Least Squares

When constraints are imposed on the factor matrix, then (2.29) is to be modified as follows (for $d \in \{1, 2, 3\}$):

$$\boldsymbol{H}_d = \underset{\boldsymbol{H}_d}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{T}_{(d)} - \boldsymbol{W}_d \boldsymbol{H}_d^\mathsf{T}\|_F^2 + r(\boldsymbol{H}_d) \tag{2.32}$$

where $r(\boldsymbol{H}_d)$ represents the constraints or regularization terms imposed on $\boldsymbol{H}_d$.

In NCP decomposition (and NMF for $N = 2$), all quantities are nonnegative. For instance, vector $\boldsymbol{a}_r$ belongs to the nonnegative orthant $\mathbb{R}_+^{I_1}$. In iterative algorithms such as ALS, this constraint is ensured at each iteration by *projecting* a computed value onto the nonnegative orthant. Accordingly, each factor matrix can be projected onto the nonnegative orthant right after its calculation; this is the NALS algorithm [25, p.47]. The pseudo-code is given in Algorithm 2 page 47, where we note that $\boldsymbol{\Lambda}_{\boldsymbol{H}_d}$ is an $R \times R$ diagonal matrix whose diagonal entries are the normalization factors of the columns of $\boldsymbol{H}_d$, and $\boldsymbol{\Lambda}$ is an Nth-order diagonal tensor of dimensions $R \times \cdots \times R$ whose entries are the diagonal of $\prod_{d=1}^N \boldsymbol{\Lambda}_{\boldsymbol{H}_d}$.

2.4.5.3 Projected-Compressed Alternating Least Squares

Sometimes, tensors can be large, thus computing CP decomposition through ALS in the presence of constraints that require projection (such as nonnegativity) can be very demanding. In this case, it would be better to compress the dimensions of the tensor first so that CP decomposition becomes accessible. Consequently, the main updates of ALS (2.30) are done using the compressed tensor $\boldsymbol{\mathcal{G}}$ and factor matrices $\boldsymbol{H}_{d_c}$ (the subscript c stands for *compressed*), then the projection is done on the decompressed factor matrices $\boldsymbol{H}_d$ that represent the original subspaces, and so on until convergence or a stopping criterion is met. The algorithm is then

referred to as "Projected-Compressed ALS" (ProCo ALS) [27]. The rest of the section is demonstrated for third-order tensors.

The compression is done through HOSVD as explained in Section 2.4.1:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{G}} \underset{1}{\bullet} \boldsymbol{V}_1 \underset{2}{\bullet} \boldsymbol{V}_2 \underset{3}{\bullet} \boldsymbol{V}_3 \tag{2.33}$$

where $\{\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3\}$ can be seen as the transitions between the decompressed and compressed modes (or subspaces) of the tensors. After that, CP decomposition is carried out on $\boldsymbol{\mathcal{G}}$ such that:

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\Lambda} \underset{1}{\bullet} \boldsymbol{H}_{1_c} \underset{2}{\bullet} \boldsymbol{H}_{2_c} \underset{3}{\bullet} \boldsymbol{H}_{3_c} \tag{2.34}$$

Combining expressions (2.33) and (2.34), we can write:

$$\begin{aligned} \boldsymbol{\mathcal{T}} &= \boldsymbol{\Lambda} \underset{1}{\bullet} (\boldsymbol{V}_1 \boldsymbol{H}_{1_c}) \underset{2}{\bullet} (\boldsymbol{V}_2 \boldsymbol{H}_{2_c}) \underset{3}{\bullet} (\boldsymbol{V}_3 \boldsymbol{H}_{3_c}) \\ &= \boldsymbol{\Lambda} \underset{1}{\bullet} \boldsymbol{H}_1 \underset{2}{\bullet} \boldsymbol{H}_2 \underset{3}{\bullet} \boldsymbol{H}_3 \end{aligned} \tag{2.35}$$

Algorithm 3 page 48 describes ProCo ALS in two steps, the pre-processing compression using HOSVD, and the computation of CP decomposition in the presence of projection-based constraints (nonnegativity).

2.4.5.4 Alternating Optimization - Alternating Direction Method of Multipliers

Alternating Optimization - Alternating Direction Method of Multipliers (AO-ADMM) was recently introduced in [53] as a powerful algorithm to compute CP decomposition, and a flexible one with constraints. Without constraints, AO-ADMM is equivalent to ALS by alternating over the factor matrix sub-problems (from where the "AO" part comes). However, when constraints are imposed, each sub-problem is solved using an ADMM approach, which makes it possible to accommodate multiple and different kinds of constraints separately for each factor matrix, such as nonnegativity, l_1-regularization (sparsity), simplex constraints, etc. In the following, we describe the proposal and its algorithm where LS is the adopted loss criterion.

Starting with the sub-problem of a factor matrix $\boldsymbol{H}_d$ subjected to constraints $r(\boldsymbol{H}_d)$:

$$\boldsymbol{H}_d = \underset{\boldsymbol{H}_d}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{T}_{(d)} - \boldsymbol{W}_d \boldsymbol{H}_d^{\mathsf{T}}\|_F^2 + r(\boldsymbol{H}_d) \tag{2.36}$$

which can also be written as:

$$\begin{aligned} \boldsymbol{H}_d = \underset{\boldsymbol{H}_d}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{T}_{(d)} - \boldsymbol{W}_d \tilde{\boldsymbol{H}}_d\|_F^2 + r(\boldsymbol{H}_d) \\ \text{s.t. } \tilde{\boldsymbol{H}}_d = \boldsymbol{H}_d^{\mathsf{T}} \end{aligned} \tag{2.37}$$

Here, ADMM separates the sub-problem into two main perspectives: that of the main CP decomposition update where $\tilde{\boldsymbol{H}}_d$ is involved such that $f(\tilde{\boldsymbol{H}}_d) = \frac{1}{2} \|\boldsymbol{T}_{(d)} - \boldsymbol{W}_d \tilde{\boldsymbol{H}}_d\|_F^2$, and that

of the constraints represented by $r(\boldsymbol{H}_d)$ where $\boldsymbol{H}_d$ is involved. Accordingly, through ADMM, expression (2.37) is separated into the following updates:

$$\begin{cases} \tilde{\boldsymbol{H}}_d \leftarrow \underset{\tilde{\boldsymbol{H}}_d}{\operatorname{argmin}}\ f(\tilde{\boldsymbol{H}}_d) + \frac{1}{2}\|\boldsymbol{H}_d^{\mathsf{T}} - \tilde{\boldsymbol{H}}_d + \boldsymbol{U}^{\mathsf{T}}\|_F^2 \\ \boldsymbol{H}_d \leftarrow \underset{\boldsymbol{H}_d}{\operatorname{argmin}}\ r(\boldsymbol{H}_d) + \frac{1}{2}\|\boldsymbol{H}_d - \tilde{\boldsymbol{H}}_d^{\mathsf{T}} + \boldsymbol{U}\|_F^2 \\ \boldsymbol{U} \leftarrow \boldsymbol{U} + \boldsymbol{H}_d - \tilde{\boldsymbol{H}}_d^{\mathsf{T}} \end{cases} \tag{2.38}$$

where $\boldsymbol{U}$ is the dual variable that measures the closeness between $\tilde{\boldsymbol{H}}_d$ and $\boldsymbol{H}_d$ at each iteration, and ADMM iterates over the three updates until a stopping criterion is met depending on $\boldsymbol{U}$ or the number of iterations. Since the update of $\tilde{\boldsymbol{H}}_d$ has a constant expression independent of the constraints, it becomes:

$$\tilde{\boldsymbol{H}}_d \leftarrow (\boldsymbol{W}_d^{\mathsf{T}}\boldsymbol{W}_d + \rho\boldsymbol{I})^{-1}(\boldsymbol{W}_d^{\mathsf{T}}\boldsymbol{T}_{(d)} + \rho(\boldsymbol{H}_d + \boldsymbol{U})^{\mathsf{T}}) \tag{2.39}$$

The pseudo-code of AO-ADMM is described in Algorithm 5 page 49 for one ADMM sub-problem, and Algorithm 4 page 49 for the AO framework using the LS loss criterion.

In [75], the authors report two issues in AO-ADMM related to the algorithm requiring matrix inversion, and its difficulty to deal with the overlapping of a simple regularization function and a linear operator, for which the algorithm **Alternating Optimization - Primal Dual Splitting (AO-PDS)** is proposed, which deals with the sub-problems using PDS instead of ADMM. We refer to [75] for more details.

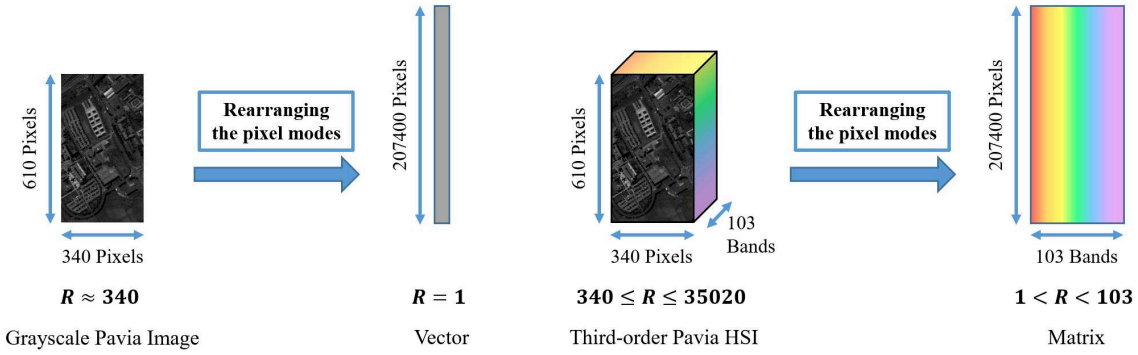
2.5 Applications in Image Analysis

This section represents an account of some applications in image analysis that involve matrix and tensor representations, with a focus on the latter. The selected applications can be put into different categories: (a) Hyperspectral Image Scene Classification, (b) Spectral Unmixing, (c) Multispectral and Hyperspectral Data Fusion, (d) Hyperspectral Image Compressive Sensing Reconstruction (HSI-CSR). In general, the selected applications deal with spectral-spatial analysis of hyperspectral images, that is to say that the spatial information is incorporated with the spectral information, of which two surveys can be found in [88, 107]. In the following sections, we expand on the first three categories. As for HSI-CSR, we only refer here to the two works using Tucker Decomposition [63] and TR decomposition [18].

Additionally, we note that CP decomposition has been considered in a real-life application of monitoring the vitality of agriculture crops by extracting optical characteristics and spatial structures of the leaves from images that are acquired by UAV-mounted cameras [98, 100, 101, 99]. In this application, we have been involved in the acquisition of images and partially in the problem statement. Technically speaking, the goal is to study the inclination direction of the leaves based on a physical relationship between the normal direction of the leaf surface, the sunlight direction, and the spectral signature of the leaf (we refer to [100] for details on

the physical relationship). Accordingly, the crops are considered healthy if the leaves stand more vertical. Moreover, the images of the leaves are acquired over different time stamps, thus we have multiple image tensors each with spectral and temporal modes, then CP or $(L_r, L_r, 1)$-decomposition would be used to extract the inclination vector coordinates. In fact, the different image tensors share some information like the direction of sunlight and the spectral reflectance of the leaves (considering we have different photos of the same type of crops), so we are facing a coupled CP or $(L_r, L_r, 1)$-decomposition problem [100, 109, 92].

At this stage, it is important to talk about some practical notes regarding the tensor rank of image-tensors. Mainly, some applications tend to rearrange the modes of pixels in lexicographic order. This procedure significantly drops the rank of the data set and allows to have a low-rank decomposition especially for applications where the arrangement or positioning of pixels is not very important and the number of extracted components is preferably low. To demonstrate why the rank drops, we give an example (see Figure 2.13) starting from a grayscale image (i.e., an image matrix), then we consider a tensor formed from multiple grayscale images as slices noting that the tensor rank of said tensor is lower bounded by the highest matrix rank among its individual grayscale images.



(a) Matrix to vector (b) Tensor to matrix

Figure 2.13: Changes in the matrix and tensor ranks upon rearranging the modes of pixels.

First, a real-life grayscale image generally has a high rank due to the complicated spatial details that appear in its representation as a matrix of real numbers, which can be seen in the example given in Figure 2.3 of a 270×130 grayscale image. In this example, through SVD, we see that reconstructing the image using a lower-rank representation requires around $R' = 80$ components (out of 130 in total) in order to maintain most of the details, which is relatively very high. Then, if a third-order tensor is formed of multiple image slices including such an image (e.g., a MSI or a HSI), the tensor rank must be higher than 80 since said tensor at least contains the structures of the example image in addition to other structures found in the other image slices. On the other hand, when the modes of pixels of a grayscale image are rearranged, we obtain a vector representation, which is rank-1 (Figure 2.13a). Then, in the case of a third-order MSI or HSI for instance, we obtain a matrix whose rank is lower than the minimum of the two dimensions, which is essentially that of the spectral mode (Figure 2.13b). Moreover, the spectral mode is often very correlated, so the rank of such a matrix would be very low as can be seen in the example given in Figure 2.4, where the spectral feature space

dropped from 103 to 4 using PCA. Figure 2.13 gives an example of the drop in the rank using the dimensions of the HSI of Pavia University.

2.5.1 Image Classification

Hyperspectral imagery refers to the acquisition of images of a scene over a wide and almost continuous spectrum, amounting to hundreds of spectral bands, to form at the end what is called a hypercube [87]. Hyperspectral imagery is a tool employed in many fields of applications such as remote sensing, monitoring Earth resources, planetary observation, biomedical imagery, non destructive testing, minearology, etc [3, 54, 1, 72, 77, 2, 45]. Due to its significance of exploring the spectral properties of a spatial setting, hyperspectral imagery has become a wide area of research with a lot of explored and potential usages, and *image classification* of the scene is one of its most common.

At its core, image classification is the problem of assigning each pixel in the image a semantic label, which at the end allows the identification of materials in the scene into recognisable properties or classes, which can be shown distinctively in a *classification map* (Figure 2.14). This kind of procedure is called *pixel-wise classification*, a scenario where generally two main variables are required: the pixels as *samples* to be classified, and the distinctive *features* (e.g., spectral response or neighborhood information) as characteristics to be classified upon. Moreover, the samples-to-features relationship can be described in a matrix form. Usually, approaching image classification using matrix or tensor decomposition comes in the form of dimensionality reduction (DR) of the feature space with PCA at the core of it.

2.5.1.1 Matrix-based and other techniques

Naturally a HSI acquires the spectral features of the scenes, mapping each pixel position, represented by a couple of positive integers (x, y), to a vector of positive real values that correspond to the material's spectral response measured by the sensor, resulting in a structure that is often represented as a data cube of third-order. When the pixels are rearranged in lexicographic order as samples, the spectral mode can then represent the set of features, making it a direct way to classify a HSI. However, due to the high spectral resolution of HSIs, pixels are considered to be in a very high dimensional feature space, which calls for DR. One conventional way to solve the problem of dimensionality is through PCA [55], then the matrix with reduced feature space is passed to the classifier [83].

In fact, HSI acquisition may contain mixed pixels, presenting challenges in acquiring the exact spectral features of the pixels and limiting the classification accuracy. As a result, the classification map may look noisy. Moreover, thanks to the advancement in optical sensor technology, spatial information surrounding the pixels is becoming more relevant especially when it comes to identifying relatively small objects. For this reason, it has become an important topic to incorporate spatial information extracted from the pixel's neighborhood in the classification, such that each pixel has features based on both spatial and spectral data

[8, 43].

Classifying images based on spatial and spectral features is called *spectral-spatial classification*. Many of such works rely on tools in the context of *Mathematical Morphology* (MM) [85, 86, 73], which is one of the most popular ways to account to spatial information. Figure 2.14 shows an illustration of classical spatial-spectral classification using MM. Tools in MM incorporate spatial information of the pixel's neighborhood, obtained directly using successive morphological transformations of an image, and stacks them as pixel-wise spatial features. Examples of popular spatial features based on MM are the Morphological Profiles (MP) [7, 6, 97] and Attribute Profiles (AP) [31, 33, 4] depending on the type of morphological transformations that are carried out. For multivariate images (as in the case of HSIs), such profiles are called Extended MP (EMP) and Extended AP (EAP), generally known to be computed using one of two strategies; the conventional marginal strategy where the profiles of the single grayscale image slices are computed separately then stacked in one way [6, 33], and the vectorial processing strategy where the extended profile is computed by processing the different image slices simultaneously [4]. If multiple kinds of attributes are considered at once, the structure is referred to as Extended Multi-Attribute Profiles (EMAP) [31]. It is worth noting here that HSI classification techniques that are purely defined in MM may end up in a very high-dimensional feature space caused by the stacking of spectral and spatial features along the same way, so they are usually combined with DR techniques (such as PCA) in the form of pre-processing and, in some cases, post-processing [71].

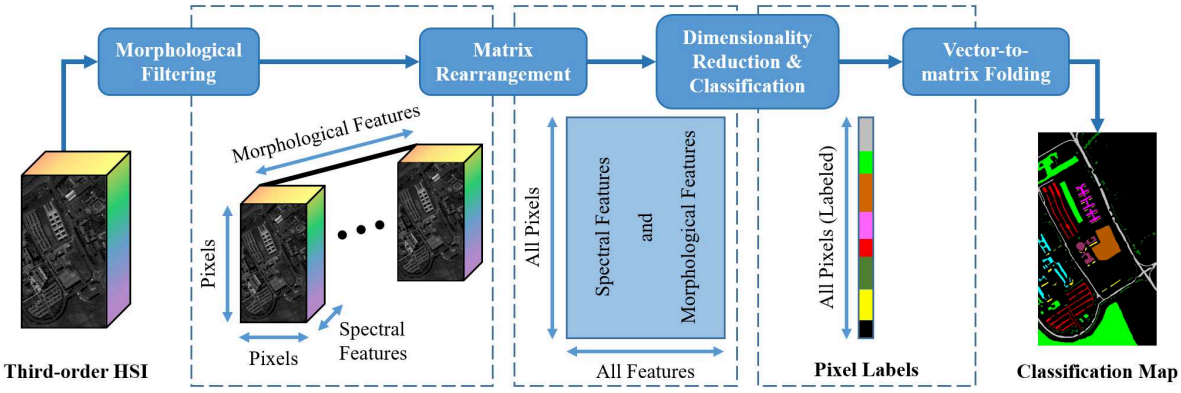


Figure 2.14: Flow chart of classical approaches for image classification incorporating mathematical morphology

While not related to matrices, and in the context of spectral-spatial classification, some works approach the problem of classification using *Composite Kernel* methods [17], by extracting spectral and spatial features and separately learning them through what would be a spectral kernel and a spatial one. Other works include contextual extraction and labeling through *Markov Random Fields* [96]. Finally, many works have been developed based on *deep learning* methods [20, 19], where principal spectral information and spatial neighborhood patches around the pixels are distributed, processed, and learned over different deep neural network architectures (e.g., Applying Encoder and Convolutional Neural Networks). Usually, deep learning methods provide very high and significant performances.

One issue regarding the mentioned methods is that, even though they prove to improve significantly on HSI feature classification with respect to classification of only spectral features, they either tend to treat the different kinds of features as one entity such as stacking spectral and spatial features as one, or they provide separate treatment of the features but do not take into account exploring the natural high-order properties and relationships among the modes of the tensors that could be obtained. As mentioned earlier in this chapter, tensors can show interesting relationships between the modes, including those among the features themselves, and these relationships are lost otherwise.

2.5.1.2 Tensor Techniques

Tensor analysis has been increasing in popularity recently especially for applications that deal with such multi-modal data [105, 102, 109], and it is still under development. By keeping the features separated in different modes (into what we would call a higher-order arrangement of the features compared to when they are stacked along the same mode), the ordering of the data and the intrinsic connections are respected, which is dealt with through *tensor decomposition*.

The first work in terms of *tensor modeling* of HSIs for the sake of supervised classification is [105], but especially recently, the application has been visited a few times [56, 58, 48, 57, 44]. In [105], a so-called Tensor-PCA (TPCA) is used as a tensor extension of PCA following the concept of HOSVD. First, the tensor is built by adding a fourth-mode in the HSI using MM through Additive Morphological Decomposition (AMD) (see Section 4.2.3.2), then the first two modes of pixels are rearranged as one. Second, TPCA is used for post-processing DR of each of the feature modes in a similar fashion to the rough description of HOSVD visualized in figure 2.7. However, when it comes to the classification phase, it still boils down to rearranging the data (i.e., the compressed tensor) into matrix form and merging the feature modes.

In order to avoid breaking the tensor structure at the classification phase, our works [56, 58, 57] were proposed to approach the problem differently using CP decomposition. The main difference is that CP decomposition directly provides a samples-to-features matrix ready to be classified upon, which is the *first factor matrix*, regardless of how many feature diversities the tensor has. Accordingly, each row of the matrix represents a pixel, and the columns represent the new extracted features that are defined by the patterns found in the other factor matrices. This work is a contribution which we present in further details in Chapter 4.

On another note, [48] approaches spatial-spectral classification by mixing many techniques distributed on several stages, of which CP decomposition plays a role in extracting rank-1 spectral and spatial features. The study also includes MM, kernel learning of the extracted features, and Support Vector Machines. This work produces significantly good results, but the procedure can be long and complex, so we refer to [48] for full details.

Interestingly, some links are being developed recently between tensor representations and machine learning [90, 66]. In the context of hyperspectral image analysis, the work of [70] proposes a Rank-R Feedforward Neural Network model for hyperspectral data analysis whose weights follow the property of rank-R CP decomposition. The advantages of this work are (a)

learning the data in multi-modal format where the weights follow a Rank-R CP decomposition property, and (b) the significant reduction in the model parameters for training thanks to this property.

2.5.2 Spectral Unmixing

One of the important advantages of HSI acquisition is to understand the spectral composition of the materials in the scene. In the process of acquisition, each pixel represents a spectral reflectance vector that reflects the physical properties of that pixel area according to the spectral range of the hyperspectral sensor. As such, some pixel areas cover one type of physical material, thus referred to as "pure pixels", while others may cover more than one material, thus referred to as "mixed pixels". Thanks to the availability of spectral dictionaries that define the spectral signatures of some pure materials, it is possible to study the spectral composition of each pixel in the scene in terms of pure materials. This application is called *spectral unmixing* (SU), which normally refers to the BSS of HSI data into a set of *endmebembers* (spectral signatures of pure materials) and *fractional abundances* (the proportions of endmembers in each pixel) [10] based on the *spectral features* that are present in the image. Figure 2.15 shows an example of a HSI with pure and mixed pixels and some elements of a SU of an urban scene.

There are many ways to approach SU problems [10], but here we focus on approaches related to the Linear Mixing Model (LMM) with matrix and tensor decomposition involved. Furthermore, many SU works tend to incorporate spatial information in them whether as pre-processing, post-processing, or as part of the unmixing, and such a survey is found in [107].

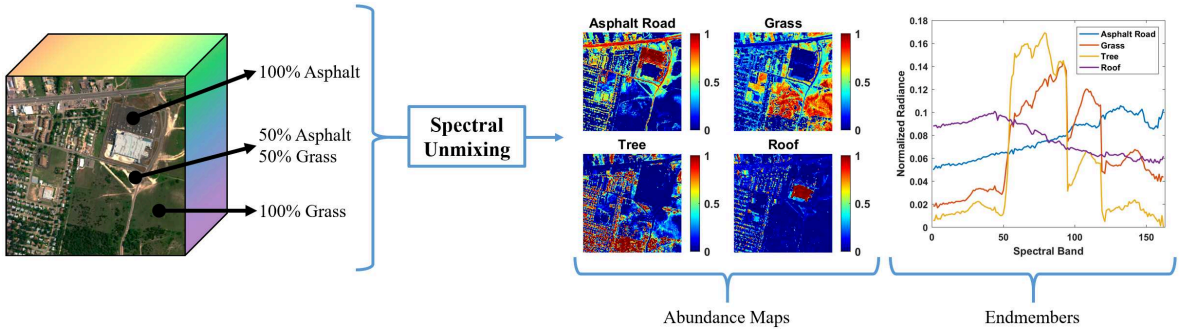


Figure 2.15: The elements of spectral unmixing. Endmembers and corresponding abundance maps are extracted from a hyperspectral image.

2.5.2.1 Matrix Techniques

LMM is one of the most popular ways to model SU problems [64]. For a HSI matrix $\boldsymbol{M}$ formed of I pixels (as rows) and J spectral bands (as columns), LMM assumes that each pixel

row $\boldsymbol{m}_i \; \forall \, i \in \{1, \ldots, I\}$ can be expressed as the linear combination of R endmembers $\boldsymbol{b}_r$:

$$\boldsymbol{m}_i = \sum_{r=1}^{R} a_{ir} \boldsymbol{b}_r \tag{2.40}$$

such that all the values are nonnegative and $\sum_{r=1}^{R} a_{ir} = 1 \; \forall \, i \in \{1, \ldots, I\}$, where the values of a_{ir} stand for the fractional abundances of the endmembers. This model is rearranged into a NMF problem in the form of $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}$ subjected to the mentioned constraints. Correspondingly, SU can be carried out by finding the two nonnegative matrices $\boldsymbol{A} \in \mathbb{R}^{I \times R}$, whose rows represent the fractional abundances at each pixel, and $\boldsymbol{B} \in \mathbb{R}^{J \times R}$, whose columns represent the extracted endmembers. Nonnegativity of the factors ensures a part-based additive representation since fractional abundances and spectral signatures are nonnegative, and the condition $\sum_{r=1}^{R} a_{ir} = 1$ is called the Abundance Sum-to-one Constraint (ASC), which ensures that the endmember composition sums to one for each pixel so that the pixels lie inside a simplex formed by the endmembers.

In general, NMF-based approaches for LMM have some problems. For instance, solution uniqueness is not guaranteed since the space of solutions may be large, and the fractional abundances may not represent a good estimation of the mixtures. For that, many constrained extensions of NMF were proposed in order to reduce the space of solutions such as [111, 82, 115, 110, 74]. Among the constrained extensions of NMF, *sparsity* constraints play a very important role in SU [111, 82, 115]. Physically speaking, it is assumed that a pixel is a mixture of only a few of the extracted endmembers, and sparsity controls the fractional abundances to reflect this assumption. Moreover, sparsity promotes solution uniqueness in the presence of nonnegative constraints. In [111, 82], some sparseness measures are proposed. In [115], a structured sparse method is proposed by utilizing spatial information in a graph scheme, where each pixel is represented as a node, and the similarity measure between adjacent pixels depends on their spectral signatures.

Spectral unmixing has also been carried out in the framework of MSI and HSI data fusion using Coupled NMF (CNMF) [113]. CNMF unmixes the data through alternating NMF between the HSI (high spectral resolution) and MSI (high spatial resolution) in order to extract the spectral signatures and the high-resolution abundance maps respectively, which enhances the quality of the unmixed components.

In fact, while LMM is seen as a direct linear model for SU, it is considered *practically insufficient* in cases where the HSI contain spectral variabilities in the form of illumination conditions or nonlinear effects. For that, *Extended LMM* (ELMM) [39] was proposed by introducing a pixel-dependent function $\boldsymbol{f}_i : \mathbb{R}^J \to \mathbb{R}^J$, which maps each endmember $\boldsymbol{b}_r$ to another spectral signature $\boldsymbol{b}_r^{(i)}$ that best reflects the targeted variabilities:

$$\boldsymbol{m}_i = \sum_{r=1}^{R} a_{ir} \boldsymbol{f}_i(\boldsymbol{b}_r) = \sum_{r=1}^{R} a_{ir} \boldsymbol{b}_r^{(i)}. \tag{2.41}$$

Dealing with spectral variability in SU using ELMM is an active topic that has seen a lot of progress [104, 42, 40, 41, 103], and we talk about it in more details in Chapter 5. Currently,

the approaches can be put in three categories based on how the spectral variability function is modeled [39]: (a) Spectral Bundles, (b) Computational Models, and (c) Parametric Models. These models are usually incorporated in a constrained matrix and tensor decomposition framework.

2.5.2.2 Tensor Approaches

However, recent SU works proposed to represent the HSI data as tensors, where NMF is insufficient to model the problem. In general, these works can be put in two categories:

- Those where the HSI is unmixed in its third-order form (without rearranging the modes of pixels) such as the work of super-resolution [61] and that of spatial smoothing [108]. In this category, information of *spatial positioning* are important, which is ignored when the pixel modes are rearranged to fit a matrix-based model.
- Those where the HSI is essentially high-order even after rearranging the pixel modes, i.e., where a pixel holds more than just the spectral diversity and where decomposing the extra diversities is relevant, such as the work of HSI in time-series [102], that of patch-HSI [103], and that of multi-feature unmixing [Chapter 5, unpublished yet]. In this category, an important advantage is having a *low-rank* decomposition.

Starting with the first category, in [61], a SRI is unmixed through a coupled CP decomposition that alternates beween a MSI and a HSI without rearranging the pixel modes (as an extension of CNMF that does not ignore the spatial positioning of the pixels). This work benefits from conserving the structure of the data and guaranteeing *identifiability*, but it has some issues related to the tensor rank assumption (see Section 2.5.3).

In [108], smoothness is imposed on the spatial modes of the image, which would be very difficult to do if the pixel modes are rearranged. We also note that this work uses $(L_r, L_r, 1)$-*decomposition* for SU, which means that the high-rankness of the tensor can be absorbed in the partitions $\boldsymbol{A}_r$ and $\boldsymbol{B}_r$, while R remains relatively low.

As for the second category, in [102], tensor-based SU using CP decomposition is first introduced, where the goal is to extract the spectral signatures and their abundances from HSI in time series. The tensor is then formed of three modes: pixels, spectral bands, and time stamps. In addition to the low-rankness, having time features guarantees uniqueness in addition to observing temporal patterns of the extracted materials.

In [103], a new mode is built by stacking relatively small neighborhood patches at the level of each pixel in an attempt to impose a neighborhood-based spatial regularization on the abundances while maintaining a low-rank CP decomposition. Moreover, the model resembles a regulated version of ELMM (RegELMM) that we describe in more details in Chapter 5.

In Chapter 5, we propose a general study and define a common framework for multi-feature hyperspectral unmixing using CP decomposition that extends the works of [102, 103, 56, 58, 57]

and emphasizes on the importance of considering artificially constructed diversities, such as MM, in the process of unmixing. At its core, we focus on the relationship of RegELMM that is described in [103] and develop over its properties using physical, graphical, and visual analysis and comparisons, and we are interested in exploring additional diversities obtained through sequential filtering of the HSI. In particular, this allows to have a relatively low-rank decomposition that is *unique* as explained for the second category, deals with the problem of *spectral variability* by considering objects that contain pixels connected by similar spectral and morphological properties (smoothing the variabilities between close spectral-spatial structures in the data) instead of patches that can contain pixels of different materials, and *jointly* extracts materials along with their spectral-morphological multi-feature patterns. Moreover, when we are not interested in the natural evolution of the materials (such as time-series), it can be enough to acquire one HSI of the scene to have a unique CP decomposition.

2.5.3 MSI and HSI Super-Resolution Data Fusion

In general, multispectral imaging sensors have high spatial resolution and low spectral one, while hyperspectral imaging sensors have low spatial resolution and high spectral one. Hyperspectral super-resolution aims at finding a super-resolution image (SRI) through a data fusion between a MSI and a HSI in order to improve the spatial resolution of HSI. A comparative review of some state-of-the-art methods can be found in [112]. As presented in [112], many methods have been proposed to solve this problem in the framework of matrix factorization such as the work of CNMF [113] that is briefly described in Section 2.5.2.1. Moreover, the same problem has been approached recently using CP decomposition and Tucker [61, 79].

In general, it is assumed that the MSI is a spectrally degraded version of the SRI, while the HSI is a spatially degraded one, by means of a set of matrices that are seen as the *degradation factors*. In the following we describe the case of CNMF, then we talk about that of CP decomposition and Tucker.

2.5.3.1 Matrix Case

We denote the matricized versions of the third-order MSI, HSI, and SRI by $\boldsymbol{M}$, $\boldsymbol{H}$, and $\boldsymbol{S}$ respectively, where their first modes represent the pixels and the second ones represent the spectral bands. We also assume that the matrices are decomposed through NMF such that $\boldsymbol{M} = \boldsymbol{A}_M \boldsymbol{B}_M^\mathsf{T}$, $\boldsymbol{H} = \boldsymbol{A}_H \boldsymbol{B}_H^\mathsf{T}$, and $\boldsymbol{S} = \boldsymbol{A}_M \boldsymbol{B}_H^\mathsf{T}$, where $\boldsymbol{A}_M$ and $\boldsymbol{A}_H$ represent the fractional abundances, and $\boldsymbol{B}_M$ and $\boldsymbol{B}_H$ represent the spectral signatures. Regarding the degradation factors, we assume a degradation matrix (spatial-wise) $\boldsymbol{X}$ such that $\boldsymbol{H} = \boldsymbol{X}\boldsymbol{S}$, and another one (spectral-wise) $\boldsymbol{Y}$ such that $\boldsymbol{M} = \boldsymbol{S}\boldsymbol{Y}^\mathsf{T}$. Figure 2.16 shows an illustration of the elements of this approach.

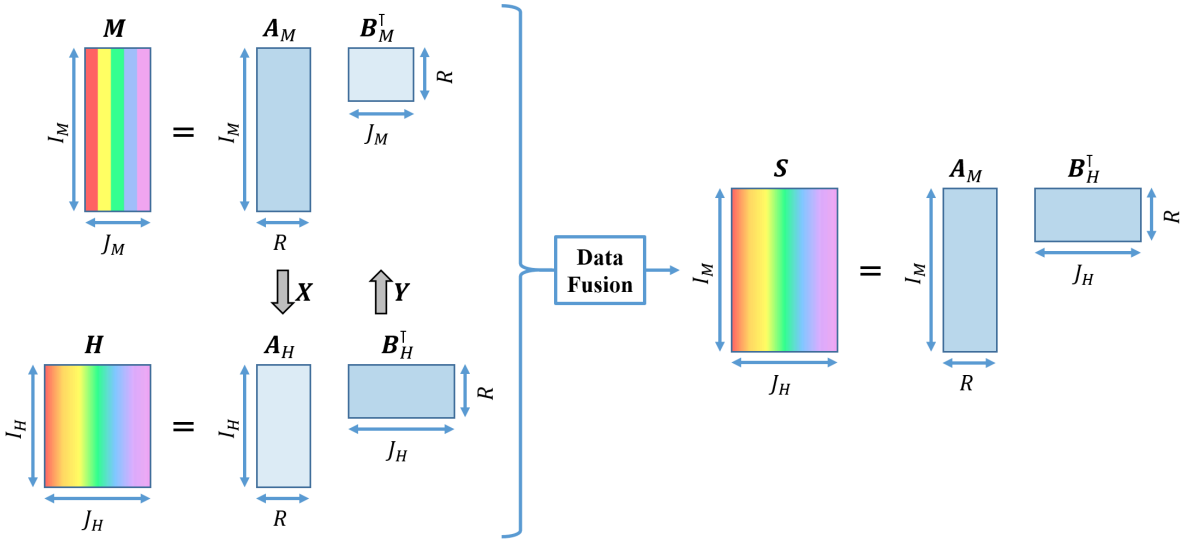


Figure 2.16: The elements of data fusion between a MSI ($\boldsymbol{M}$) and a HSI ($\boldsymbol{H}$) in order to obtain a SRI ($\boldsymbol{S}$). The figure is inspired by the one used in [113].

2.5.3.2 Tensor Case

Recently, two works were proposed using tensor decomposition in order to take advantage of the natural third-order structure of the data for more accurate results [61, 79]. The main difference between works that are based on matrix factorization and those that are based on tensor decomposition is that in the latter, we have three degradation factors, one for each of the modes.

The work of [61] proposes to solve the problem using CP decomposition. In addition to processing the images in their natural third-order forms, CP decomposition guarantees the identifiability of the SRI. However, this work assumes that the SRI admits a low-rank CP decomposition, which can be a risky assumption because such tensors naturally have very high tensor ranks.

In order to deal with the problem of low-rankness assumption, a Tucker HOSVD-based approach is proposed [79], which assumes that the tensor has a low multi-linear rank, especially from the perspective of the spectral-mode unfolding (i.e., SVD($\boldsymbol{T}_{(3)}$); for a tensor $\boldsymbol{\mathcal{T}}$ whose third mode represents the spectra). Even though the Tucker model is not generally identifiable, the SRI is proved to be identifiable for a range of multi-linear ranks.

2.6 Concluding Remarks

Through this chapter, we provided descriptions and references in the literature for matrix and tensor decomposition techniques that are relevant to image analysis, for anyone who is interested in the subject, especially that it has potential to grow. We started with some

basic definitions, described the different matrix and tensor decompositions and their properties (ranks, uniqueness, algorithms), then gave an account on different applications of image analysis where tensor decomposition was involved.

In the following chapters, we talk about the contributions of this thesis. The first one is rather algorithmic (Chapter 3), while the others are more application-oriented. Moreover, the focus is going to shift rather to CP decomposition and hyperspectral imagery as the former was the main tool that was used in the contributions and the latter was the main subject of interest in the applications.

Algorithm 2 Nonnegative Alternating Least Squares (NALS)

Require: $\boldsymbol{\mathcal{T}}$, initialize $\boldsymbol{H}_2^{(0)}$, initialize $\boldsymbol{H}_3^{(0)}$
  $t = 0$;
  while Stopping criterion is not met, **do**
    $t \leftarrow t + 1$
    Update the factor matrices:
    $\boldsymbol{H}_1^{(t)} \leftarrow \boldsymbol{T}_{(1)} \left[(\boldsymbol{H}_3^{(t-1)} \odot \boldsymbol{H}_2^{(t-1)})^{\mathsf{T}} \right]^{\dagger}$;
    $\boldsymbol{H}_1^{(t)} \leftarrow \max(0, \boldsymbol{H}_1^{(t)})$
    $\boldsymbol{H}_2^{(t)} \leftarrow \boldsymbol{T}_{(2)} \left[(\boldsymbol{H}_3^{(t-1)} \odot \boldsymbol{H}_1^{(t)})^{\mathsf{T}} \right]^{\dagger}$;
    $\boldsymbol{H}_2^{(t)} \leftarrow \max(0, \boldsymbol{H}_2^{(t)})$
    $\boldsymbol{H}_3^{(t)} \leftarrow \boldsymbol{T}_{(3)} \left[(\boldsymbol{H}_2^{(t)} \odot \boldsymbol{H}_1^{(t)})^{\mathsf{T}} \right]^{\dagger}$;
    $\boldsymbol{H}_3^{(t)} \leftarrow \max(0, \boldsymbol{H}_3^{(t)})$
    Let $\boldsymbol{H}_3^{(t)}$ absorb the column-normalization factors of $\boldsymbol{H}_1^{(t)}$ and $\boldsymbol{H}_2^{(t)}$:
    $\boldsymbol{H}_1^{(t)} \leftarrow \boldsymbol{H}_1^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{H}_1}^{-1}$;
    $\boldsymbol{H}_2^{(t)} \leftarrow \boldsymbol{H}_2^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{H}_2}^{-1}$;
    $\boldsymbol{H}_3^{(t)} \leftarrow \boldsymbol{H}_3^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{H}_1} \boldsymbol{\Lambda}_{\boldsymbol{H}_2}$;
  end while
  Normalize the columns of $\boldsymbol{H}_3^{(t)}$: $\boldsymbol{H}_3^{(t)} \leftarrow \boldsymbol{H}_3^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{H}_3}^{-1}$
  Reshape the product $\boldsymbol{\Lambda}_{\boldsymbol{H}_1} \boldsymbol{\Lambda}_{\boldsymbol{H}_2} \boldsymbol{\Lambda}_{\boldsymbol{H}_3}$ into a third-order diagonal tensor $\boldsymbol{\Lambda}$
  return $\boldsymbol{H}_1^{(t)}$, $\boldsymbol{H}_2^{(t)}$, $\boldsymbol{H}_3^{(t)}$, $\boldsymbol{\Lambda}$

Algorithm 3 ProCo ALS

Require: $\boldsymbol{\mathcal{T}}$, $\boldsymbol{dim_c}$, initialize $\boldsymbol{H}_2^{(0)}$, initialize $\boldsymbol{H}_3^{(0)}$
Step 1 - Compress $\boldsymbol{\mathcal{T}}$:
for $d \in \{1, 2, 3\}$ **do**
Compute $\boldsymbol{V}_d$ from the SVD of the d-mode unfolding of $\boldsymbol{\mathcal{T}}$ such that $\boldsymbol{T}_{(d)} = \boldsymbol{U}_d \boldsymbol{\Sigma}_d \boldsymbol{V}_d$;
$\boldsymbol{V}_d \leftarrow$ Take out the last $\boldsymbol{dim_c}(d)$ columns of $\boldsymbol{V}_d$;
end for
$\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{T}} \bullet_1 \boldsymbol{V}_1^\mathsf{T} \bullet_2 \boldsymbol{V}_2^\mathsf{T} \bullet_3 \boldsymbol{V}_3^\mathsf{T}$;
$\boldsymbol{H}_{2_c}^{(0)} \leftarrow \boldsymbol{V}_2^\mathsf{T} \boldsymbol{H}_2^{(0)}$;
$\boldsymbol{H}_{3_c}^{(0)} \leftarrow \boldsymbol{V}_3^\mathsf{T} \boldsymbol{H}_3^{(0)}$;
—
Step 2 - Compute CP decomposition with constraints:
repeat
for $d \in \{1, 2, 3\}$ **do**
$\boldsymbol{W}_d = \odot_{j \neq d} \boldsymbol{H}_{j_c}$;
$\boldsymbol{H}_{d_c} \leftarrow \boldsymbol{G}_{(d)} \left(\boldsymbol{W}_d^\mathsf{T}\right)^\dagger$; (Factor update)
$\boldsymbol{H}_d \leftarrow \boldsymbol{V}_d \boldsymbol{H}_{d_c}$; (Decompression)
$\boldsymbol{H}_d \leftarrow \max(0, \boldsymbol{H}_d)$; (Projection)
$\boldsymbol{H}_{d_c} \leftarrow \boldsymbol{V}_d^\mathsf{T} \boldsymbol{H}_d$; (Compression)
end for
Let $\boldsymbol{H}_{3_c}$ absorb the column-normalization factors of $\boldsymbol{H}_{1_c}$ and $\boldsymbol{H}_{2_c}$:
$\boldsymbol{H}_{1_c} \leftarrow \boldsymbol{H}_{1_c} \boldsymbol{\Lambda}_{\boldsymbol{H}_{1_c}}^{-1}$;
$\boldsymbol{H}_{2_c} \leftarrow \boldsymbol{H}_{2_c} \boldsymbol{\Lambda}_{\boldsymbol{H}_{2_c}}^{-1}$;
$\boldsymbol{H}_{3_c} \leftarrow \boldsymbol{H}_{3_c} \boldsymbol{\Lambda}_{\boldsymbol{H}_{1_c}} \boldsymbol{\Lambda}_{\boldsymbol{H}_{2_c}}$;
until A stopping criterion is met.
Normalize the columns of $\boldsymbol{H}_{3_c}$: $\boldsymbol{H}_{3_c} \leftarrow \boldsymbol{H}_{3_c} \boldsymbol{\Lambda}_{\boldsymbol{H}_{3_c}}^{-1}$
Reshape the product $\boldsymbol{\Lambda}_{\boldsymbol{H}_{1_c}} \boldsymbol{\Lambda}_{\boldsymbol{H}_{2_c}} \boldsymbol{\Lambda}_{\boldsymbol{H}_{3_c}}$ into a third-order diagonal tensor $\boldsymbol{\Lambda}$
return $\boldsymbol{\Lambda}$, $\boldsymbol{H}_1 = \boldsymbol{V}_1 \boldsymbol{H}_{1_c}$, $\boldsymbol{H}_2 = \boldsymbol{V}_2 \boldsymbol{H}_{2_c}$, $\boldsymbol{H}_3 = \boldsymbol{V}_3 \boldsymbol{H}_{3_c}$

Algorithm 4 Alternating Optimization (Least Squares loss)

Require: $\boldsymbol{\mathcal{T}}$, $\boldsymbol{H}_1$,$\boldsymbol{H}_2$,$\boldsymbol{H}_3$, $\boldsymbol{U}_1$,$\boldsymbol{U}_2$,$\boldsymbol{U}_3$
  Initialize $\{\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_3\}$;
  Initialize $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \boldsymbol{U}_3\}$ to zero matrices;
  for $d \in \{1, 2, 3\}$ **do**
    Store the d-mode unfolding $\boldsymbol{T}_{(d)}$;
  end for
  repeat
    for $d \in \{1, 2, 3\}$ **do**
      $\boldsymbol{W}_d = \odot_{j \neq d} \boldsymbol{H}_j$;
      Update $\boldsymbol{H}_d$ and $\boldsymbol{U}_d$ using Algorithm 5;
    end for
    Update μ if necessary; (refer to [53] for the update of μ)
  until A termination criterion is reached (number of iterations)
  Normalize the columns of the factor matrices and store the weights in $\boldsymbol{\Lambda}$;
  return $\boldsymbol{\Lambda}$, $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, $\boldsymbol{H}_3$

Algorithm 5 ADMM of a fixed mode d

Require: $\boldsymbol{T}$, $\boldsymbol{W}$, $\boldsymbol{H}$, $\boldsymbol{U}$, R, μ, ϵ, i_{max}
  $\boldsymbol{G} = \boldsymbol{W}^\mathsf{T}\boldsymbol{W}$;
  $\rho = \text{trace}(\boldsymbol{G})/R$;
  Calculate $\boldsymbol{L}$ from Cholesky decomposition such that: $\boldsymbol{G} + (\rho + \mu)\boldsymbol{I_k} = \boldsymbol{L}\boldsymbol{L}^\mathsf{T}$;
  $\boldsymbol{F} = \boldsymbol{W}^\mathsf{T}\boldsymbol{T}$;
  $\boldsymbol{H_f} = \boldsymbol{H}$;
  repeat
    $\tilde{\boldsymbol{H}} \leftarrow (\boldsymbol{L}^\mathsf{T})^{-1}\boldsymbol{L}^{-1}(\boldsymbol{F} + \rho(\boldsymbol{H} + \boldsymbol{U})^\mathsf{T} + \mu\boldsymbol{H_f}^\mathsf{T})$; (See [53] for μ and $\boldsymbol{H_f}$)
    $\boldsymbol{H} \leftarrow \text{argmin}_{\boldsymbol{H}}\, r(\boldsymbol{H}) + \frac{1}{2}\|\boldsymbol{H} - \tilde{\boldsymbol{H}}^\mathsf{T} + \boldsymbol{U}\|_F^2$;
    $\boldsymbol{U} \leftarrow \boldsymbol{U} + \boldsymbol{H} - \tilde{\boldsymbol{H}}^\mathsf{T}$;
    Update r and s; (refer to [53] for the updates of r and s)
  until ($r < \epsilon$ and (s is undefined or $s < \epsilon$)) or ($imax > 0$ and $i \geq imax$)
  return $\boldsymbol{H}$ and $\boldsymbol{U}$

Chapter 3

Modified NALS

Sommaire

3.1 Introduction

Tensor decompositions are still in the process of study and development. In this chapter, we talk about an algorithmic contribution of this thesis [60], in which we point out a problem existing in nonnegative tensor decompositions, stemming from the representation of decomposable tensors by outer products of vectors, and propose approaches to solve it. In fact, a scaling indeterminacy appears whereas it is not inherent in the decomposition, and the choice of scaling factors has an impact during the execution of iterative algorithms and should not be overlooked. Computer experiments support the interest in the greedy algorithm proposed, in the case of the CP decomposition.

Here, we refer to the introduction of tensors in Section 2.4. In particular, we shall focus our attention on the CP decomposition of third-order tensors and the ALS algorithm, which have been described in Sections 2.4.2 and 2.4.5. As for the rest of this chapter, Section 3.2 states our motivation. Section 3.3 shows the proposed approach. In Section 3.4, we show a toy example of the problem and how it can be solved with the proposed approach. After that, we discuss the computer results in Section 3.5. Finally, we draw some concluding remarks.

3.2 Motivation

Nonnegativity. As explained in Section 2.4.2.3, when the observation tensor $\boldsymbol{\mathcal{T}}$ contains only real nonnegative entries, it is suitable to impose decomposable tensors $\boldsymbol{\mathcal{D}}_r$ to also be nonnegative, which is important for applications in image analysis. By doing this, we denote the nonnegative rank by R^+, which may be larger than R. It is thus necessary for the rest of the chapter to denote the nonnegative CP decomposition of a nonnegative tensor as:

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R^+} \lambda_r \, \boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r, \tag{3.1}$$

where $a_{ir} \in \mathbb{R}^+$, $b_{jr} \in \mathbb{R}^+$ and $c_{kr} \in \mathbb{R}^+$, $\forall(i,j,k,r)$. It is also important to recall that, given any real tensor $\boldsymbol{\mathcal{T}}$ of rank R, it is fortunately always possible to find a best nonnegative approximation of $\boldsymbol{\mathcal{T}}$ of given nonnegative rank R^+ [69, 81].

Projection onto the nonnegative orthant: In the NCP decomposition (3.1), all quantities are nonnegative. In iterative algorithms such as NALS [25] and ProCo ALS [27], this constraint is ensured at each iteration by projecting a computed value onto the nonnegative orthant (as explained in Sections 2.4.5.2 and 2.4.5.3). This is where the problem of projection-induced erroneous computations shows up. In fact, projecting $\boldsymbol{\mathcal{D}}_r$ or its building vectors $\{\boldsymbol{a}_r, \boldsymbol{b}_r, \boldsymbol{c}_r\}$ do not yield the same result. Since this observation is already true for matrices, a simple example will be most convincing.

Example. Take a matrix $\boldsymbol{H}$ below, of rank 1. Now its projection $\boldsymbol{H}^+ = \max(0, \boldsymbol{H})$ has rank 2. So it is preferred to project its supporting vectors $\{\boldsymbol{a}, \boldsymbol{b}\}$ instead. The obtained vectors are $\{\boldsymbol{a}^+, \boldsymbol{b}^+\}$ and yield a matrix of nonnegative rank equal to 1:

$$\boldsymbol{H} = \begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \otimes \begin{pmatrix} 2 \\ -1 \end{pmatrix} = \boldsymbol{a} \otimes \boldsymbol{b},$$

$$\boldsymbol{H}^+ = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \text{ whereas } \boldsymbol{a}^+ \otimes \boldsymbol{b}^+ = \begin{pmatrix} 4 & 0 \\ 0 & 0 \end{pmatrix}.$$

The problem is that vectors $\{\boldsymbol{a}, \boldsymbol{b}\}$ are not uniquely defined. We could have taken $\{-\boldsymbol{a}, -\boldsymbol{b}\}$ without changing $\boldsymbol{H}$. Should we do that, we obtain instead:

$$\boldsymbol{H} = \begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \boldsymbol{a} \otimes \boldsymbol{b},$$

$$\boldsymbol{H}^+ = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}, \text{ whereas } \boldsymbol{a}^+ \otimes \boldsymbol{b}^+ = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

We see that the projected matrix of nonnegative rank 1 is not the same. This issue comes from the fact that no care has been taken of the scaling indeterminacies (which reduce to sign indeterminacies thanks to the use of factors λ_r) inherent in the representation of a rank-1 tensor by a triplet of vectors.

NALS. Among the procedures that resort to projection, hard thresholding is the procedure in which it is the easiest to illustrate the occurring of the problem. One simple algorithm that has been widely used to compute NCP decomposition is the NALS algorithm [25, p.47], which is explained in Section 2.4.5.2. NALS minimizes the following cost function Φ with respect to the nonnegative factor matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ in an alternating fasion:

$$\Phi = \sum_{ijk} (t_{ijk} - \sum_{r=1}^{R} \lambda_r \, a_{ir} \, b_{jr} \, c_{kr})^2. \tag{3.2}$$

where each factor matrix can be projected onto the nonnegative orthant right after its update at each iteration. The pseudo-code is given in Algorithm 2 page 47.

3.3 Proposed Approach

We illustrate the problem with hard thresholding (cf. Section 3.4), but our solution could also reveal useful in soft thresholding as well. The problem is worse when all entries in a column vector are set to zero; this prevents its normalization (as it would lead to a division by zero) or imposes an erroneous reduction of the rank (due to the arbitrary removal of the null columns). The solution we describe overcomes these two difficulties most of the time, up to negligible extraneous computation load. We propose to implement this in a procedure to be executed before projection. The concept goes as follows. Because of normalization, the scaling indeterminacy in $\boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r$ reduces merely to signs. In fact, in every decomposable tensor $\boldsymbol{\mathcal{D}}_r$, we have two variables, ϵ, $\eta \in \{-1, +1\}$, which are to be used as sign flippers for the columns $\boldsymbol{a}_r$, $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ that are together involved in an outer product term, without changing the result of the outer product given by:

$$\begin{aligned} \boldsymbol{\mathcal{D}}_r = \boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r = (\epsilon \, \eta \, \boldsymbol{a}_r) \otimes (\epsilon \, \boldsymbol{b}_r) \otimes (\eta \, \boldsymbol{c}_r) = \boldsymbol{a}'_r \otimes \boldsymbol{b}'_r \otimes \boldsymbol{c}'_r \\ \forall (\epsilon, \eta) \in \{-1, +1\}. \end{aligned} \tag{3.3}$$

Expression (3.3) covers all 4 combinations of sign flipping of vectors, without affecting the result of the original outer product. Now denote by $\boldsymbol{a}'_r = \epsilon \, \eta \, \boldsymbol{a}_r$, $\boldsymbol{b}'_r = \epsilon \, \boldsymbol{b}_r$, and $\boldsymbol{c}'_r = \eta \, \boldsymbol{c}_r$, and:

$$\begin{cases} \boldsymbol{a}_r^- = \max(0, -\boldsymbol{a}_r), & \boldsymbol{a}_r^+ = \max(0, \boldsymbol{a}_r) \\ \boldsymbol{b}_r^- = \max(0, -\boldsymbol{b}_r), & \boldsymbol{b}_r^+ = \max(0, \boldsymbol{b}_r) \\ \boldsymbol{c}_r^- = \max(0, -\boldsymbol{c}_r), & \boldsymbol{c}_r^+ = \max(0, \boldsymbol{c}_r) \end{cases} \tag{3.4}$$

In particular, $\boldsymbol{a}_r^+ - \boldsymbol{a}_r^- = \boldsymbol{a}_r$.

Given a triplet of vectors, $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}\}$, there are 4 possibilities to construct a nonnegative decomposable tensor $\boldsymbol{\mathcal{D}}^{[\ell]}$ $\forall \, \ell \in \{1, 2, 3, 4\}$ by just flipping their signs (the values of ϵ and η) without changing the result of the outer product $\boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c}$ and by setting negative values to zero in $\{\boldsymbol{a}', \boldsymbol{b}', \boldsymbol{c}'\}$ (refer to Table 3.1).

(ϵ,η)	$(+1,+1)$	$(-1,-1)$	$(+1,-1)$	$(-1,+1)$
ℓ	1	2	3	4
$\boldsymbol{\mathcal{D}}^{[\ell]}$	$\boldsymbol{\mathcal{D}}^{[1]} = \boldsymbol{a}^+ \otimes \boldsymbol{b}^+ \otimes \boldsymbol{c}^+$	$\boldsymbol{\mathcal{D}}^{[2]} = \boldsymbol{a}^+ \otimes \boldsymbol{b}^- \otimes \boldsymbol{c}^-$	$\boldsymbol{\mathcal{D}}^{[3]} = \boldsymbol{a}^- \otimes \boldsymbol{b}^+ \otimes \boldsymbol{c}^-$	$\boldsymbol{\mathcal{D}}^{[4]} = \boldsymbol{a}^- \otimes \boldsymbol{b}^- \otimes \boldsymbol{c}^+$

Table 3.1: The 4 possibilities created by sign flipping. For the sake of convenience, $\boldsymbol{a}^+$ here stands for vector $(\boldsymbol{a}^{[\ell]})^+ = \max(0, \boldsymbol{a}^{[\ell]})$, and $\boldsymbol{a}^-$ stands for vector $(\boldsymbol{a}^{[\ell]})^- = \max(0, -\boldsymbol{a}^{[\ell]})$ which applies similarly for $\boldsymbol{b}^+$, $\boldsymbol{b}^-$, $\boldsymbol{c}^+$ and $\boldsymbol{c}^-$.

We are interested to know which combination would yield the minimal number of resets. Ultimately, we are concerned about (a) avoiding to set a whole vector to zero, which would lead to decrease the rank. This goal can mean "set as few entries to zero as possible". And we also aim at (b) minimizing the distance between the original tensor and its nonnegative approximation.

We explored several criteria. The first is to minimize the cost function:

$$\Phi_0(\ell) = \|\boldsymbol{\mathcal{T}} - \sum_r \boldsymbol{\mathcal{D}}_r^{[\ell]}\|_F^2 \tag{3.5}$$

This criterion is very costly to optimize, due to the large number of combinations. In fact, for every r, there are four possibilities to assign (ϵ, η), and this assignment can be different for each r. This would result in 4^R possibilities to explore. This is why we propose two greedy algorithms searching for the optimal solution $\boldsymbol{\mathcal{D}}_r^{[\ell]}$ independently for every r. One possibility is to minimize with respect to ℓ the following product **for every** r **independently**, and for the L^2 norm:

$$\Phi_1(\ell, r) = \|\boldsymbol{\mathcal{D}}_r - \boldsymbol{\mathcal{D}}_r^{[\ell]}\|_F^2. \tag{3.6}$$

Let us express this criterion for $\ell = 1$, without loss of generality. For any fixed r, we have:

$$\Phi_1(1, r) = \|\boldsymbol{\mathcal{D}}_r\|_F^2 + \|\boldsymbol{\mathcal{D}}_r^{[1]}\|_F^2 - 2\sum_{ijk} a_{ir} a_{ir}^+ b_{jr} b_{jr}^+ c_{kr} c_{kr}^+. \tag{3.7}$$

The last term can be rewritten as $2\,(\boldsymbol{a}_r^\mathsf{T} \boldsymbol{a}_r^+)(\boldsymbol{b}_r^\mathsf{T} \boldsymbol{b}_r^+)(\boldsymbol{c}_r^\mathsf{T} \boldsymbol{c}_r^+)$. Next, it is also equal to $2\,\|\boldsymbol{a}_r^+\|_F^2\,\|\boldsymbol{b}_r^+\|_F^2\,\|\boldsymbol{c}_r^+\|_F^2$, since $\boldsymbol{a}_r^+$ and $\boldsymbol{a}_r^-$ are orthogonal and $\boldsymbol{a}_r = \boldsymbol{a}_r^+ - \boldsymbol{a}_r^-$. This suggests another criterion to minimize w.r.t. ℓ:

$$\Phi_2(\ell, r) = \|\boldsymbol{a}_r^-\|_F^2 \cdot \|\boldsymbol{b}_r^-\|_F^2 \cdot \|\boldsymbol{c}_r^-\|_F^2 \tag{3.8}$$

Criteria Φ_1 and Φ_2 are easy to optimize with respect to (ϵ, η), *i.e* with respect to ℓ, and need negligible extraneous computation load. Algorithm 6 describes the pseudo-code of minimizing Φ_1, and Algorithm 7 describes the modification in NALS (Modified NALS), which can be compared to NALS in Algorithm 2 page 47.

Algorithm 6 Minimization of Φ_1 w.r.t. to ϵ and η for fixed r

Require: $\boldsymbol{a}_r$, $\boldsymbol{b}_r$, $\boldsymbol{c}_r$ (all real-valued)
  initialize $\epsilon(\ell)$ and $\eta(\ell)$ for each possible $\boldsymbol{\mathcal{D}}^{[\ell]}$; (according to Table 3.1)
  normalize: $\boldsymbol{a} \leftarrow \frac{\boldsymbol{a}_r}{\|\boldsymbol{a}_r\|}$; $\boldsymbol{b} \leftarrow \frac{\boldsymbol{b}_r}{\|\boldsymbol{b}_r\|}$; $\boldsymbol{c} \leftarrow \frac{\boldsymbol{c}_r}{\|\boldsymbol{c}_r\|}$;
  compute $\boldsymbol{\mathcal{D}} = \boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c}$;
  $\ell = 0$;
  while $\ell \leq 4$, **do**
    $\ell = \ell + 1$;
    compute $\boldsymbol{\mathcal{D}}^{[\ell]}$ based on Table 3.1;
    $\Phi_1(\ell) \leftarrow \|\boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{D}}^{[\ell]}\|_F^2$;
  end while
  Find $\ell_o = \operatorname{argmin}_\ell \Phi_1(\ell)$;
  $\boldsymbol{a}_r \leftarrow \epsilon(\ell_o)\,\eta(\ell_o)\,\boldsymbol{a}_r$; $\boldsymbol{b}_r \leftarrow \epsilon(\ell_o)\,\boldsymbol{b}_r$; $\boldsymbol{c}_r \leftarrow \eta(\ell_o)\,\boldsymbol{c}_r$;
  return $\boldsymbol{a}_r$, $\boldsymbol{b}_r$, $\boldsymbol{c}_r$;

3.4 Toy Example

In the following, we show a toy example of a third-order rank-2 tensor CP decomposition where the problem occurs (Section 3.4.1), and how the proposed approach deals with it (Section 3.4.2). Consider the following factor matrices from which the simulated data tensor is formed, i.e. $\boldsymbol{\mathcal{T}} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}) \cdot \boldsymbol{\Lambda}$:

$$\boldsymbol{A} = \begin{bmatrix} 0.8025 & 0.1914 \\ 0.0089 & 0.9106 \\ 0.5966 & 0.3662 \end{bmatrix}, \; \boldsymbol{B} = \begin{bmatrix} 0.0088 & 0.7495 \\ 1 & 0.6620 \end{bmatrix}, \; \boldsymbol{C} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0.7071 & 0 \\ 0.7071 & 0 \end{bmatrix}.$$

This case guarantees uniqueness, so decomposing $\boldsymbol{\mathcal{T}}$ using rank $R = 2$ should give back the same matrices. When computing the CP Decomposition with Algorithm 2, after one update of $\boldsymbol{A}$, one of its columns becomes negative, and hence that column gets discarded at the end as being "Undefined", and the rank of the approximation is decreased by 1 even though, prior to that, the rank would be exact. Note that, for the sake of conciseness, during the loop of updates, only the columns of $\boldsymbol{A}$ and $\boldsymbol{B}$ are normalized and their norms are absorbed in the columns of $\boldsymbol{C}$; after the loop ends, $\boldsymbol{C}$ is normalized and its column norms (containing those of $\boldsymbol{A}$ and $\boldsymbol{B}$) form the values of $\boldsymbol{\Lambda}$ (see Algorithm 2).

3.4.1 Standard NALS

$$\boldsymbol{A}: \begin{bmatrix} 0.2311 & -0.0464 \\ 0.1891 & -0.0627 \\ 0.2178 & -0.0498 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2311 & 0 \\ 0.1891 & 0 \\ 0.2178 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.6252 & \text{Undefined} \\ 0.5118 & \text{Undefined} \\ 0.5893 & \text{Undefined} \end{bmatrix} \rightarrow \begin{bmatrix} 0.6252 \\ 0.5118 \\ 0.5893 \end{bmatrix}$$

Algorithm 7 Modified NALS

Require: $\boldsymbol{\mathcal{T}}$, initialize $\boldsymbol{B}^{(0)}$, initialize $\boldsymbol{C}^{(0)}$
  $t = 0$;
  while Stopping criterion is not met, **do**
    $t \leftarrow t + 1$
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{T}_{(1)} \left[(\boldsymbol{C}^{(t-1)} \odot \boldsymbol{B}^{(t-1)})^{\mathsf{T}} \right]^{\dagger}$;
    $\boldsymbol{B}^{(t)} \leftarrow \boldsymbol{T}_{(2)} \left[(\boldsymbol{C}^{(t-1)} \odot \boldsymbol{A}^{(t)})^{\mathsf{T}} \right]^{\dagger}$;
    $\boldsymbol{C}^{(t)} \leftarrow \boldsymbol{T}_{(3)} \left[(\boldsymbol{B}^{(t)} \odot \boldsymbol{A}^{(t)})^{\mathsf{T}} \right]^{\dagger}$;
    $r = 0$;
    while $r < R$ **do**
      Update $\boldsymbol{a}_r$, $\boldsymbol{b}_r$, and $\boldsymbol{c}_r$ using Algorithm 6;
      $r \leftarrow r + 1$;
    end while
    $\boldsymbol{A}^{(t)} \leftarrow \max(0, \boldsymbol{A}^{(t)})$; $\boldsymbol{B}^{(t)} \leftarrow \max(0, \boldsymbol{B}^{(t)})$; $\boldsymbol{C}^{(t)} \leftarrow \max(0, \boldsymbol{C}^{(t)})$;
    Let $\boldsymbol{C}^{(t)}$ absorb the column-normalization factors of $\boldsymbol{A}^{(t)}$ and $\boldsymbol{B}^{(t)}$:
    $\boldsymbol{A}^{(t)} \leftarrow \boldsymbol{A}^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{A}}^{-1}$; $\boldsymbol{B}^{(t)} \leftarrow \boldsymbol{B}^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{B}}^{-1}$; $\boldsymbol{C}^{(t)} \leftarrow \boldsymbol{C}^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{A}} \boldsymbol{\Lambda}_{\boldsymbol{B}}$;
  end while
  Normalize the columns of $\boldsymbol{C}^{(t)}$: $\boldsymbol{C}^{(t)} \leftarrow \boldsymbol{C}^{(t)} \boldsymbol{\Lambda}_{\boldsymbol{C}}^{-1}$
  return $\boldsymbol{A}^{(t)}$, $\boldsymbol{B}^{(t)}$, $\boldsymbol{C}^{(t)}$, $\boldsymbol{\Lambda}$

$$\boldsymbol{B}: \begin{bmatrix} 0.2962 & 0 \\ 1.0561 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2962 & 0 \\ 1.0561 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2701 & \text{Undefined} \\ 0.9628 & \text{Undefined} \end{bmatrix} \rightarrow \begin{bmatrix} 0.2701 \\ 0.9628 \end{bmatrix}$$

$$\boldsymbol{C}: \begin{bmatrix} 0 & 0 \\ 0.4978 & 0 \\ 0 & 0 \\ 1.3779 & 0 \\ 1.3779 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0.4978 & 0 \\ 0 & 0 \\ 1.3779 & 0 \\ 1.3779 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0.2018 & 0 \\ 0 & 0 \\ 0.5585 & 0 \\ 0.5585 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0.2018 \\ 0 \\ 0.5585 \\ 0.5585 \end{bmatrix}$$

At the end the output of Standard NALS results in the following approximated matrices with a high RMSE:

$$\boldsymbol{A} = \begin{bmatrix} 0.8004 \\ 0.0249 \\ 0.5990 \end{bmatrix}; \boldsymbol{B} = \begin{bmatrix} 0.0165 \\ 0.9999 \end{bmatrix}; \boldsymbol{C} = \begin{bmatrix} 0 \\ 0.0835 \\ 0 \\ 0.7046 \\ 0.7046 \end{bmatrix}$$

3.4.2 Modified NALS

$$\boldsymbol{A}: \begin{bmatrix} 0.2311 & -0.0464 \\ 0.1891 & -0.0627 \\ 0.2178 & -0.0498 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2311 & 0.0464 \\ 0.1891 & 0.0627 \\ 0.2178 & 0.0498 \end{bmatrix} \rightarrow \begin{bmatrix} 0.6252 & 0.5015 \\ 0.5118 & 0.6772 \\ 0.5893 & 0.5384 \end{bmatrix}$$

$$\boldsymbol{B}: \begin{bmatrix} 0.3420 & 0.4425 \\ 1.1203 & 0.6211 \end{bmatrix} \rightarrow \begin{bmatrix} 0.3420 & 0.4425 \\ 1.1203 & 0.6211 \end{bmatrix} \rightarrow \begin{bmatrix} 0.2919 & 0.5802 \\ 0.9564 & 0.8145 \end{bmatrix}$$

$$\boldsymbol{C}: \begin{bmatrix} 0 & 0 \\ -0.4978 & -7.6392 \\ 0 & 0 \\ 2.9691 & 11.1635 \\ 2.9691 & 11.1635 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 7.6392 \\ 0 & 0 \\ 2.9691 & 0 \\ 2.9691 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 \\ 0 & 0.5392 \\ 0 & 0 \\ 1.2853 & 0 \\ 1.2853 & 0 \end{bmatrix}$$

At the end the output of Modified NALS results in the following approximated matrices with zero error:

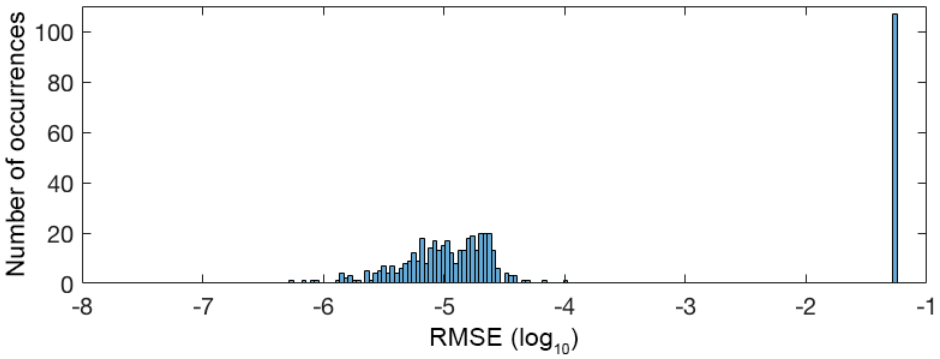
$$\boldsymbol{A} = \begin{bmatrix} 0.8025 & 0.1914 \\ 0.0089 & 0.9106 \\ 0.5966 & 0.3662 \end{bmatrix}; \boldsymbol{B} = \begin{bmatrix} 0.0088 & 0.7495 \\ 1 & 0.6620 \end{bmatrix}; \boldsymbol{C} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0.7071 & 0 \\ 0.7071 & 0 \end{bmatrix}$$

3.5 Computer Results

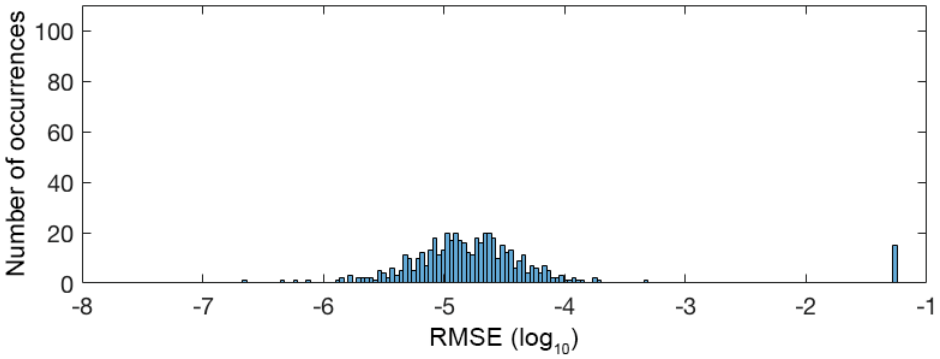
Here, we demonstrate the performance of the proposed algorithm. First, 500 realizations of 10×5 nonnegative matrices $\{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\}$ are drawn. The rank of the tensor that is tested is hence $R = 5$. Entries of factor matrices are the absolute value of i.i.d (independent and identically distributed) drawn from a standard Gaussian distribution. On each realization, both standard NALS and a modified version based on the minimization of Φ_1 are run.

As can be seen in Figure 3.1a, 107 realizations out of 500 are unsuccessful using standard NALS, that is, 107 realizations generate at least one fully negative column in a factor matrix which is then zeroed due to hard thresholding. This eventually leads to a decrease of the rank down to 4 or less and hence to a large reconstruction error in Φ_0 (close to 10^{-1}). Among those 107 pathological cases, our simple function described by Algorithm 6 could cope with 92 of them (Figure 3.1b) without a significant increase in complexity. However, 15 realizations remain unsolved because they correspond to either one of two particular cases: (a) either one column, say $\boldsymbol{a}_{r_o}$, is fully negative, and the two others, namely $\boldsymbol{b}_{r_o}$ and $\boldsymbol{c}_{r_o}$ are fully positive, or (b) all the three columns are fully negative.

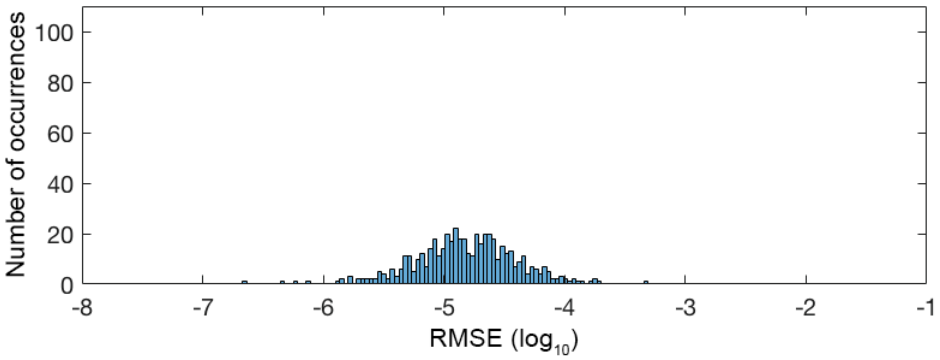
In order to cope with the latter particular cases, a straightforward improvement was brought in Algorithm 7 page 56, by drawing a fresh column vector (also as the absolute



(a) Standard NALS



(b) Modified NALS



(c) Modified NALS with column re-initialization

Figure 3.1: Histograms of the error, in log_{10} scale, obtained after 500 iterations. Top: Standard NALS. Middle: Modified NALS with cost function Φ_1. Bottom: NALS Modified with cost function Φ_1 and with column re-initialization.

value of i.i.d drawn from a standard Gaussian distribution) to replace null vectors when generated in either one of the two unsolved pathological cases, before normalizing the columns. Figure 3.1c shows that all the 15 remaining cases were solved and the rank was preserved.

Finally, the progress of the reconstruction error per iteration in log_{10} scale for each of the three cases can be seen in Figure 3.2. Here, we can see that the plots of Standard and Modified NALS (in blue and red), where some columns in the factor matrices got discarded

as "undefined", converged to high values of the reconstruction error because the rank of the approximation tensor becomes less than that of the data tensor ($R = 5$), thus the degrees of freedom are not enough to store the decomposed tensor data. On the other hand, when the rank of the approximation tensor is conserved at that of the data tensor, the decomposition is exact and yields a very low value of the reconstruction error (yellow plot).

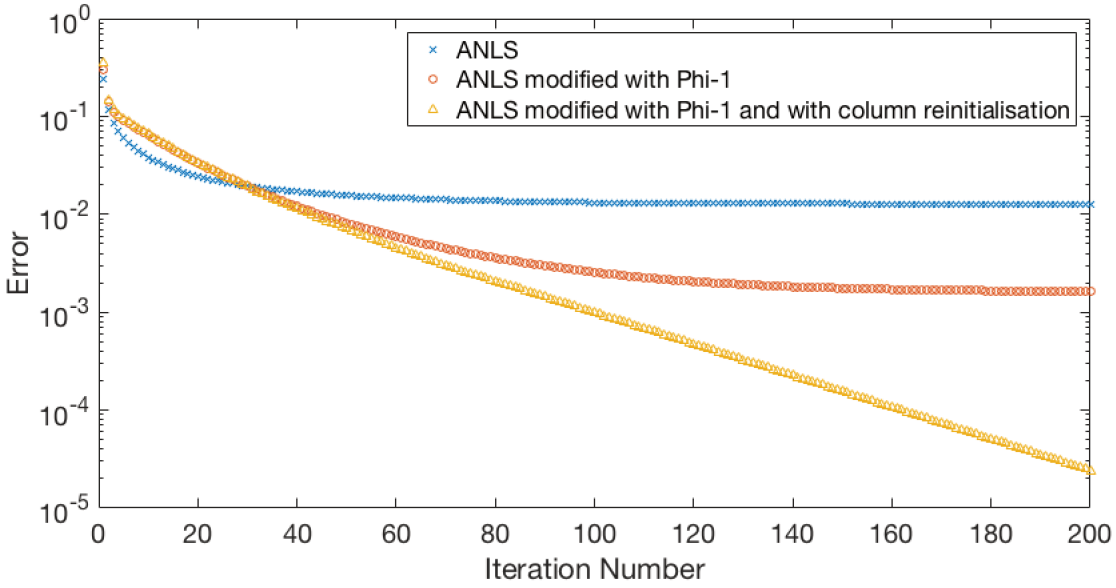


Figure 3.2: The average error obtained after 500 iterations, as a function of the number of iterations in Standard NALS (blue crosses), Modified NALS with Φ_1 (red circles), and Modified NALS with Φ_1 and with column reinitialization (yellow triangles).

3.6 Concluding Remarks

In this chapter, we have emphasized the fact that rank-1 tensors should not be treated as a collection of vectors without care, and showed an illustration in the case of Standard NALS using hard thresholding. In the latter case, two modifications have been proposed to fix the problem. In future works, one possibility is to investigate applications to other algorithms such as ADMM, and/or using soft thresholding. The influence of noise would also deserve to be further addressed.

Another possibility is to use this problem as an automatic estimation of the rank of the data tensor while decomposing the latter through CP decomposition. For instance, since the input rank of the decomposition can sometimes be overestimated, and since this value is also the number of columns of the factor matrices, one could investigate a trade-off by allowing some columns to be discarded while the reconstruction error does not converge to a high value.

This work was rather algorithmic and abstract, but it concerns nonnegative approximations of tensors using CP decomposition which indirectly involves applications of image analysis.

In fact, the problem appeared in the first place while applying NCP decomposition on tensors containing image data. In the following two chapters, we talk about two contributions that are directly involved in applications of image analysis.

Chapter 4

Hyperspectral Image Classification Based on Mathematical Morphology and Tensor Decomposition

Sommaire

Abstract

In Section 2.5.1, an overview of spectral-spatial scene classification was given with the use of Mathematical Morphology. In this chapter, this topic is explored through tensor representations and Canonical Polyadic (CP) decomposition. For instance, a HSI is a third-order data block, and building new spatial diversities may increase this order. In many cases, since pixel-wise classification requires a matrix of pixels and features, HSI data are reshaped as matrices which causes high dimensionality and ignores the multi-modal structure of the features. This chapter deals with HSI classification by modeling the data as tensors of high order. More precisely, multi-modal hyperspectral data is built and dealt with using tensor CP decomposition.

Experiments on real HSI show the effectiveness of the CP decomposition as a candidate for classification thanks to its properties of representing the pixel data in a matrix compact form with a low dimensional feature space while maintaining the multi-modality of the data.

4.1 Introduction

In Section 2.5, we gave some background on HSI scene classification and talked about some of the works that involve matrix and tensor factorization. Among these works, we mentioned those of [56, 58, 57], which are contributions of this thesis to HSI scene classification based on MM and tensor decomposition. In this chapter, we expand on the details of these contributions and show the experiments and results. The rest of this section recalls some elements from Section 2.5 and serves as an introduction of the following sections.

Image classification is the problem of assigning each pixel in the image a semantic label (or a "class"), which at the end allows the identification of materials in the scene into recognisable properties. The idea behind this problem is that pixels are identified through a set of features. A direct and natural example of this scenario is hyperspectral imaging. For instance, a HSI acquires the spectral features of the scenes, mapping each pixel to a vector of positive real values representing the spectral response measured by the sensor, which is referred to as the *spectral reflectance* of the corresponding material in the scene. This kind of framework presents pixels as *samples*, and spectral information as *features*. For that, HSI scene classification has become an interesting area of application for its potential and usefulness.

In the literature, we find that features are not restricted to spectral information. Many works attempted to explore the spatial information of the scene since neighborhood information can improve the classification accuracy of pixels [8, 43]. As explained in Section 2.5, a considerable part of research was done in exploring the representation of this pixels-to-feature(s) relationship and processing the features, which included Composite Kernel methods, Markov Random Fields, Deep Learning methods, Dimensionality Reduction (DR), etc. One of the ways to represent pixels and features is through matrices, where each row represents a sample pixel, which is often used in DR techniques and has proved to be efficient.

In fact, *the pixel-to-feature representation as a matrix* is the core of this work, which we attempt to obtain *directly* using tensor decomposition as well as in a *DR* framework, but first, and in order to avoid a lot of repetition, it is important to recall that here we incorporate spatial information through MM as briefly described in Section 2.5, using EMP, AMD, and EMAP, with a difference from classical matrix approaches that the feature modes are kept separated, such that the data are treated naturally through *tensor analysis*.

In fact, *tensor modeling* of HSI for the sake of classification has previously been visited [105, 56, 58]. In [105], TPCA is used in a similar fashion to HOSVD (and Figure 2.7) in order to obtain a tensor with reduced feature spaces. However, when it comes to the classification phase, the method still boils down to rearranging the data into matrix form and merging the

feature modes. In order to avoid breaking the tensor structure at the classification phase, one way is to approach the whole problem differently through CP decomposition (CPD) [29].

The main difference is that CPD directly provides a *samples-to-features matrix* that would be ready to pass to the classifier. This matrix is the *first factor matrix* in the CPD which is associated with the mode of pixels regardless of how many feature modes the tensor has (for instance, some experiments here are carried out with tensors having three feature modes). Accordingly, each row of said matrix represents a pixel, and the columns represent the new extracted features that can be qualitatively defined by the patterns found in the other factor matrices. Moreover, the choice of the rank in the approximation, which is also the number of extracted features in CPD, is more flexible in CPD than HOSVD, which also acts as a direct DR technique since the choice of the rank is relatively low. Through CPD, our goal is to *directly* obtain a matrix of samples and features through one tensor decomposition that:

- Intuitively accounts to the high-order relationships.
- Provides the matrix with a relatively low-dimensional feature space. This can be seen as DR where a high-order pixel representation reduces to a row of coefficients.
- Conserves the information found in the tensor given a value of the rank.

This goal was explored with preliminary results that can be found in the conference contributions ISMM 2019 [56] and IGARSS 2019 [58], and the following is an extension of those two. [56] and [58] include preliminary results using AMD and EMAP as morphological settings applied on one dataset. The current version includes more detailed analysis, better modeled results, more morphological settings, an additional dataset, and further discussions regarding the possible use of unconstrained CPD.

First, we start by creating high-order tensors from HSI using spatial transformations derived from MM, then we jointly handle the different variety of features by means of nonnegative CPD (NCPD). For example, in terms of spatial transformations, one might want to create multiple modes, each based on a certain morphological concept, which results in a block of data that is at least of order four: i.e., two for spatial dimensions (image rows and columns), one for spectral features, and another one for morphological transformations. Normally, the first two modes are rearranged in lexicographic order, which leads to a tensor of order three at least, where each pixel represents at least a matrix of features. For instance, if the tensor is of order four or higher, one pixel starts to represent a tensor of features. This cannot be directly fed to the classifier, and here comes the significance of CPD. This is partly illustrated in the first two steps of the flowchart of Figure 4.1, where $\boldsymbol{\mathcal{T}}$ is the Morphological HSI Tensor that is to be decomposed using CPD.

CPD decomposes a tensor into several factor matrices, each of which represents one diversity of the tensor. This is to say that CPD directly produces a matrix that represents the pixels, which we choose to pass to the classifier without the need to matricize the tensor by rearranging its feature modes. This is partly illustrated in the last two steps of the flowchart of Figure 4.1, where $\boldsymbol{H}_1$ is said factor matrix. CPD also enjoys some properties like uniqueness

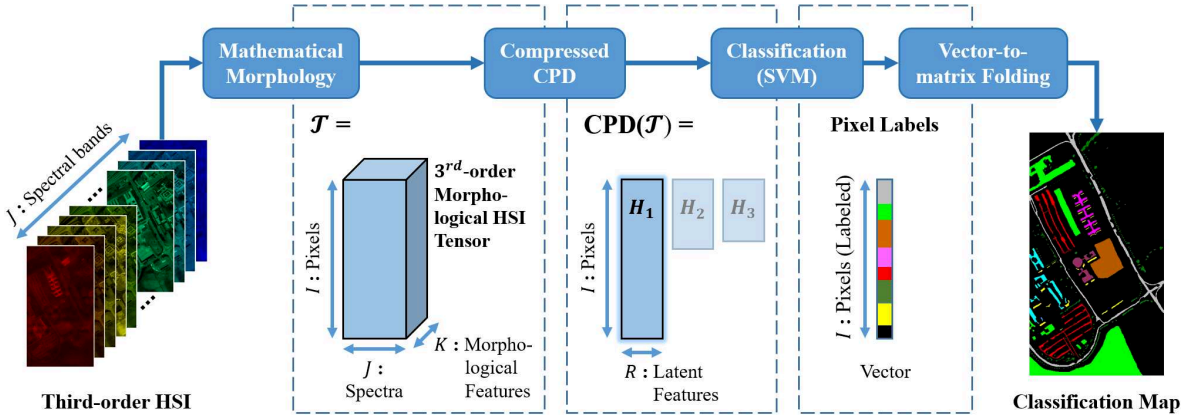


Figure 4.1: A flowchart of spectral-spatial classification using MM and CPD.

under mild conditions and flexibility with constraints such as nonnegativity. Experiments are conducted on two real HSI datasets. For each dataset, some experiments are done on third-order tensors, having one spectral mode and another spatial one, and other experiments are done on fourth-order tensors, having one spectral mode and two spatial ones. For third-order tensors, we have two kinds of experiments, the first one is based on EMP, and the second one is based on an AMD that is inspired by [105]. For fourth-order tensors, the experiments are based on an EMAP [31] formed using four different kinds of attribute transformations. We note that in [31], an EMAP was built when multiple EAP of the same HSI are concatenated along the same mode for classification, instead, we stack them as a new mode in the tensor. The experiments show promising results in improving the classification accuracy compared to other methods in the literature. Moreover, a discussion is added in the case of unconstrained CPD.

From here on, this chapter is organized as follows. Section 4.2 starts with a background on MM for image processing, then talks in Section 4.2.3 about the higher-order arrangement of MM transformations as tensors. In Section 4.3 we recall some aspects of CPD then talk about the contributions of this work. In Section 4.4 we present the experiments and discuss the results. Finally, we give some conclusion remarks in Section 4.5.

4.2 Background on Mathematical Morphology

In this section, we start by introducing the basic concepts and notions of MM as the set of transformations in image processing; we recall some definitions and properties of morphological and attribute filters that are used in this project, then we explain briefly the concept of morphological and attribute profiles. After that, we move on to talk about the case examples for building high-order multi-modal data; the EMP-based example and the ones inspired by the AMD and the EMAP methods.

4.2.1 Basics of Mathematical Morphology

In image processing, morphological operators are image-to-image neighborhood-based transformations. Let's assume that Ψ is a morphological transformation, then Ψ maps an image to another where the output value of a pixel is dependent on the values of its neighboring pixels. For a given 2D image $\boldsymbol{I}$, we define some properties of image-based transformations:

- Idempotence: $\Psi(\Psi(\boldsymbol{I})) = \Psi(\boldsymbol{I})$
- Extensivity: $\Psi(\boldsymbol{I}) \succeq \boldsymbol{I}$
- Anti-extensivity: $\Psi(\boldsymbol{I}) \preceq \boldsymbol{I}$

where the inequalities are understood entry-wise.

Such transformations are important in image processing for instance to extract features of interest in the image based on spatial characteristics. Depending on the desired spatial characteristics, we focus on two families of filters defined in the framework of MM, "morphological filters" [91] and "attribute filters" [32].

4.2.1.1 Morphological Filters Based on Structuring Elements

This family of operators is mostly used for geometrical features of the image. Such operators depend on two parameters: (1) the type of transformation and (2) the size and shape of the neighborhood, with the latter being the major key point, known as the Structuring Element (SE), which is a mask predefined shape and size (in pixels) that is meant to interact with the image and govern the interaction of pixels with their neighborhood. Suppose two SE of sizes $\mathtt{SE}_2 \subseteq \mathtt{SE}_1$, we say that the operator Ψ follows the absorption law if $\Psi_{\mathtt{SE}_1}(\Psi_{\mathtt{SE}_2}) = \Psi_{\mathtt{SE}_1}$.

For example, Erosion ($\epsilon_{\mathtt{SE}}$) and Dilation ($\delta_{\mathtt{SE}}$) are two basic morphological operators; Erosion shrinks regions with values that are greater than those of their surrounding pixels by means of the SE while Dilation dilates them. Another two types of such operators are Opening ($\gamma_{\mathtt{SE}}$) and Closing ($\phi_{\mathtt{SE}}$), Opening is the Dilation of the Erosion of an image $\boldsymbol{I}$ by means of the same SE while Closing goes in reverse. As a result, we can say that Opening removes white spaces that are smaller than the SE while Closing fills dark spaces that are smaller than the SE. In many cases, we are interested in conserving the details of objects that are larger than the SE (objects which remain unfiltered). This leads us to the family of operators by reconstruction, where an image transformation is repeated until idempotence is reached under the constraint of a mask ($\boldsymbol{M}$) that is the original image itself ($\boldsymbol{M} = \boldsymbol{I}$), we denote such transformations by $\Psi^{\infty}_{\mathtt{SE}}(\boldsymbol{M}, \boldsymbol{I})$ [106]. Opening by Reconstruction $\gamma^{\infty}_{\mathtt{SE}}(\boldsymbol{M}, \boldsymbol{I})$ and Closing by Reconstruction $\phi^{\infty}_{\mathtt{SE}}(\boldsymbol{M}, \boldsymbol{I})$ are examples of that.

We note that Opening by reconstruction and Closing by reconstruction follow the absorption law, and they are widely used for Morphological Profiles (MP), which is to be described in section 4.2.2.

4.2.1.2 Attribute Filters

Attribute filters (attribute thinning and thickening) were introduced as a new technique to extract distinctive types of spatial features, called attributes, while addressing some limitations of filters based on SE [31]. Such filters can be geometric (analogous to morphological Opening and Closing), or based on characteristics of the data distribution. The concept of attribute filtering is not based on SE, but totally on connected regions of the image sharing similar *attribute* characteristics and rearranged as component trees. In other terms, attribute filtering is defined by a criterion that evaluates the desired attribute according to a specified threshold. Any property that applies to image regions can be considered as an attribute, examples of that are the area of the regions, the moment of inertia, the standard deviation, *etc.*

Attribute thinning and thickening can be used for building Attribute Profiles (AP), which is to be described in section 4.2.2.

4.2.2 Morphological and Attribute Profiles

Before we dive into the definition of such profiles, we note that attribute thinnings and thickenings are analogous to morphological openings and closings by reconstruction respectively. In terms of properties, thinning and opening by reconstruction are anti-extensive, while thickening and closing by reconstruction are extensive, but, conversely to opening and closing, thinning and thickening are not increasing transformations [31, 33]. As such, we proceed to explain the concept of profiles considering the case of MP noting that the transition to explaining the case of AP is obvious.

For a set of extensive and anti-extensive dual operators, respectively $\{\bar{\Psi}^i, \underline{\Psi}^i\}_{i=1\ldots m}$, where m denotes the number of SEs (sometimes denoted by λ for AP), the MP of a grayscale image $\boldsymbol{I}$ can be defined as:

$$\mathrm{MP}(\boldsymbol{I}) := \{\underline{\Psi}^m(\boldsymbol{I}), \ldots, \underline{\Psi}^1(\boldsymbol{I}), \boldsymbol{I}, \bar{\Psi}^1(\boldsymbol{I}), \ldots, \bar{\Psi}^m(\boldsymbol{I})\}, \tag{4.1}$$

Figure 4.2 shows an example of a MP of a grayscale image sample. More precisely, to create a MP, these morphological transformations of the same image are stacked and rearranged along the third mode in increasing order of extensivity, that is, the elements of $\mathrm{MP}(\boldsymbol{I})$ are ordered such that $\mathrm{MP}(\boldsymbol{I})_j \prec \mathrm{MP}(\boldsymbol{I})_{j+1}$ where $\mathrm{MP}(\boldsymbol{I})_j$ refers to the j-th element of $\mathrm{MP}(\boldsymbol{I})$, with the original image stored at the middle. For multivariate images such as HSI, the concept is extended but the same idea still holds. In this case, where we focus on the marginal strategy, the profiles of each grayscale image (whether the image is a principal component or the original band itself) are concatenated along the third mode forming what is referred to in the literature as an "extended profile", which is then given as an input to the classifier. In the case of MP, extended profiles are referred to as EMP [6]. Figure 4.3 shows an illustration of conventionally creating an EMP of a multivariate image using Opening and Closing by reconstruction.

Additionally, we note that creating morphological diversity for multivariate images is not limited to extended profiles. Works in the literature extended this concept to other possible

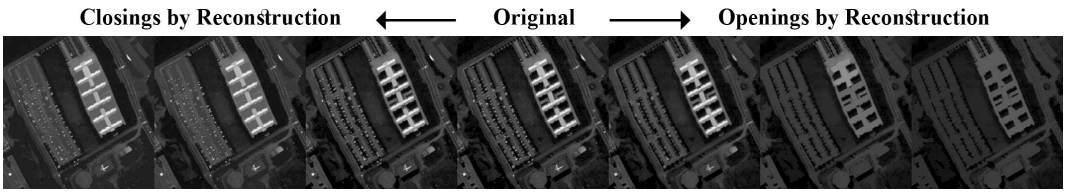


Figure 4.2: A MP using closings and openings by reconstruction of a grayscale sample from the HSI of Pavia University

rearrangements and derivations such as the Differential Morphological Profile (DMP) [7], which is created by stacking images obtained from the differences between the successive transformations and thus can be seen as the derivative of MP. Another example is AMD, which we will talk about in more detail in the next subsection.

In the case of AP, extended profiles are referred to as EAP, and, as mentioned at the beginning of this subsection, creating an EAP is analogous to that of an EMP, but using successive values of the attribute parameter. Furthermore, the work on EAP for image classification was extended to EMAP, which is to be discussed in the next subsection.

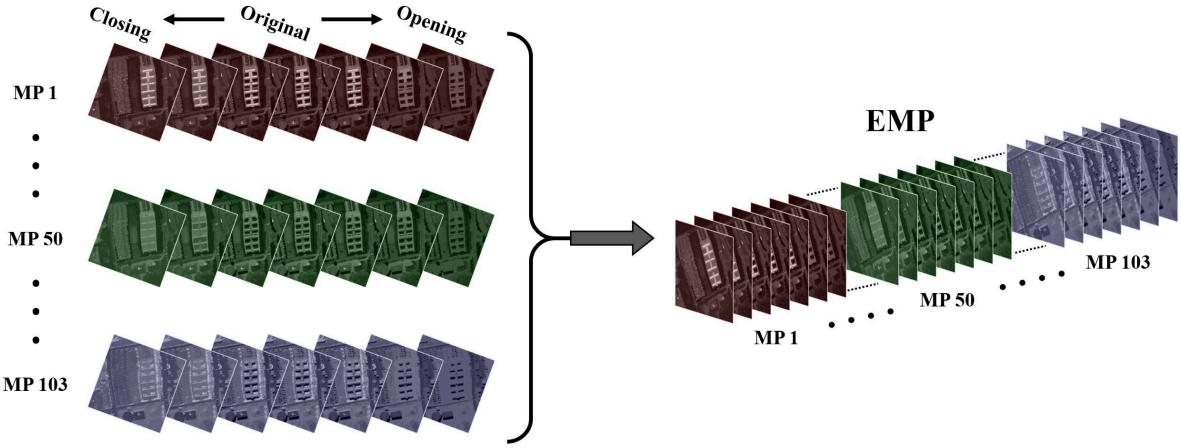


Figure 4.3: An illustration showing one way of creating an EMP of a multivariate image. On the left, it shows sample Morphological Profiles (MP) of some spectral bands, numbered by the index of the spectral band, of a cropped portion of the HSI of Pavia University. On the right, it shows the stacking of the MPs along the third mode to form the Extended Morphological Profile (EMP). The transformations were carried out using three different SE; disks with sizes [1,6,11].

4.2.3 High-Order Tensors Using MM

Having talked about the various ways to build spatial features using MM for image classification, we mention three examples that we adopt in our experiments; the first one is the EMP method which is direct and simple, the second one is a method inspired by AMD because it provides significant features and it has been previously used as a tensor model for multi-modal

feature space reduction, and the third one is inspired by EMAP as a way to create more than one spatial diversity, all of which are explained in the following.

4.2.3.1 Fourth-Order Tensor Using EMP

As mentioned earlier, the conventional way to build the EMP of a HSI is by stacking the MP of the single spectral bands along the spectral mode, forming a third-order block of data. This process considers single morphological transformations ($\Psi(\boldsymbol{I})$) only as a mapping from a 2D image to another. Instead, we choose to consider the transformation of a HSI ($\Psi(\boldsymbol{\mathcal{I}})$) as another third-order cube of data, i.e. as a mapping from a 3D image to another, under the same assumption that the transformation of a HSI is equivalent to applying the same transformation on its bands one by one. As a result, the EMP of a HSI can be seen as a higher-order MP arrangement, or in other words, a fourth-order analogy to the MP of a grayscale image, by stacking the third-order transformations of the HSI as follows:

$$\mathrm{MP}(\boldsymbol{\mathcal{I}}) := \{\underline{\Psi}^m(\boldsymbol{\mathcal{I}}), \ldots, \underline{\Psi}^1(\boldsymbol{\mathcal{I}}), \boldsymbol{\mathcal{I}}, \bar{\Psi}^1(\boldsymbol{\mathcal{I}}), \ldots, \bar{\Psi}^m(\boldsymbol{\mathcal{I}})\}, \tag{4.2}$$

The structure in (4.2) brings the same values as those found in the conventional EMP, except that the former is rearranged to separate spatial features from spectral ones. The result is a fourth-order block of data of size $I_1 \times I_2 \times J \times K$, where $I_1 \times I_2$ is the number of pixels, J is the number of spectral bands, and K is the number of morphological transformations. Figure 4.4a shows an illustration of a fourth-order tensor built through EMP.

4.2.3.2 Fourth-Order Tensor Using AMD

An AMD is formed by decomposing an image in an additive manner such that the morphological transformations sum up to give back the image itself. What is stacked in this case is not directly the original image and its transformations, but the components of the additive decomposition that are composed of one "structure component" and the "residuals". Following the same notation of $\{\bar{\Psi}^i, \underline{\Psi}^i\}_{i=1\ldots m}$ from Section 4.2.2, we give the formulas of the consecutive residual components as defined in [105]:

$$\boldsymbol{R}_i^+ = \bar{\Psi}^i(\bar{\Psi}^{i-1}(\boldsymbol{I})) - \bar{\Psi}^{i-1}(\bar{\Psi}^{i-2}(\boldsymbol{I})) \geq 0 \tag{4.3}$$

$$\boldsymbol{R}_i^- = \underline{\Psi}^{i-1}(\underline{\Psi}^{i-2}(\boldsymbol{I})) - \underline{\Psi}^i(\underline{\Psi}^{i-1}(\boldsymbol{I})) \geq 0 \tag{4.4}$$

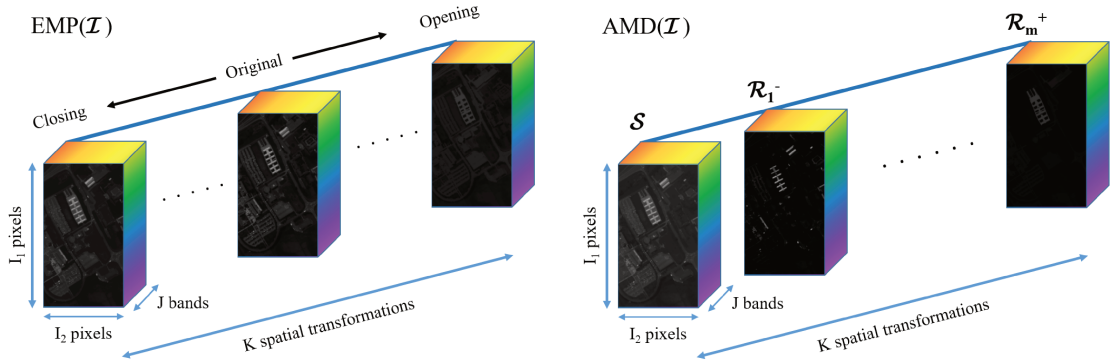
where the inequalities are understood entry-wise. The AMD of an image $\boldsymbol{I}$ can then be written as:

$$\begin{aligned} \boldsymbol{I} &= \frac{\bar{\Psi}^m(\bar{\Psi}^{m-1}(\boldsymbol{I})) + \underline{\Psi}^m(\underline{\Psi}^{m-1}(\boldsymbol{I}))}{2} + \sum_{i=1}^{m} \frac{\boldsymbol{R}_i^- - \boldsymbol{R}_i^+}{2} \\ &= \boldsymbol{S} + \sum_{i=1}^{m} \boldsymbol{R}_i, \end{aligned} \tag{4.5}$$

where the term $\boldsymbol{S}$ is considered to be the structure component because it contains the unfiltered components in the decomposition, and the terms $\{\boldsymbol{R}_i\}_{i=1\ldots m}$ are called the residuals

because they contain the residues obtained from the differences between each two successive morphological transformations. In the case of HSI, the structure and the residuals are of order three, and at the end they are stacked along the fourth mode to create a fourth-order block of data.

In our approach, we choose to build the morphological decomposition using the set $\{\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{R}}_i^-, \boldsymbol{\mathcal{R}}_i^+\}_{i=1\ldots m}$ instead of $\{\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{R}}_i\}_{i=1\ldots m}$ to preserve the nonnegativity of the data, which will be relevant for physical interpretation in the future. Thus, for the sake of consistency, we refer to it as Nonnegative-AMD (NAMD). As such, the obtained fourth-order hyperspectral data set is denoted by $\boldsymbol{\mathcal{Y}}$ and arranged in such an order: $[\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{R}}_1^-, \boldsymbol{\mathcal{R}}_1^+, \ldots, \boldsymbol{\mathcal{R}}_m^-, \boldsymbol{\mathcal{R}}_m^+]$, which is represented by an $I_1 \times I_2 \times J \times K$ array, and $K = 2m + 1$ is the number of terms along the morphological decomposition. Figure 4.4b shows an illustration of a fourth-order tensor built through EMP.



(a) EMP Tensor (b) NAMD Tensor

Figure 4.4: An illustration of fourth-order tensors using EMP and NAMD

4.2.3.3 Fifth-Order Tensor Using EMAP

Similarly to EMP, an EAP of a multivariate image is normally a third-order block of data where the spectral and spatial features are stacked along the same mode. We also choose to separate the features to form a block of data of size $I_1 \times I_2 \times J \times K$, where K is the number of attribute transformations. The concept of EMAP is to stack multiple EAP of the image so we would have a variety of spatial information for each pixel depending on different types of attributes before passing it to the classifier. In [31], EMAP was used by concatenating four kinds of EAP on a multivariate image composed of the principal components of the original HSI. We note that an EMAP is normally a third-order block of data, which means that the spectral features are mixed with the spatial ones, and that the multiple attribute profiles are mixed as well.

In our approach, we compute more than one type of EAP, each of which are of order four, then we stack them in parallel to form a fifth mode that can be seen as an acquisition of different types of EAP. The obtained fifth-order block of data is denoted by $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times J \times K \times L}$,

with L being the number of attributes taken into consideration.

4.3 Contributions

After obtaining the data set as a result of morphological transformations, we now deal with it as a tensor of high-order. In fact, each pixel now is composed of at least a matrix of features which is indeed complex. CPD is a powerful tool to break the complexity of tensors into data forms (matrices) that are easy and intuitive to deal with without the need to alter the structure of the tensor. In other words, we show that decomposing the high-order image data using CPD provides a direct and simple way to represent the pixels in matrix form that is to be classified.

In this section, we start by recalling what is relevant to this work in terms of tensor decomposition. CPD is used to decompose the tensor, but the latter can be huge and its decomposition can be computationally demanding, so we first tend to compress it using HOSVD. After that, we talk about our contributions. We first dive into the algorithmic part and our implementation of this work, then we talk about the importance of using tensor decomposition when multiple types of features are involved.

In Section 4.2.3, we saw how tensor $\mathcal{Y}$ can be of fourth and fifth order. As discussed in Section 2.5, since a 2D (grayscale) image usually has a high rank, then tensors built upon this image should be of higher rank. For that, the pixels are rearranged in lexicographic order, which reduces the rank and works better for classification since the latter requires one mode of pixels (as samples). The data tensor is then denoted by $\mathcal{T}$, with a first mode of dimension $I = I_1 \times I_2$. A fourth-order data tensor with rearranged pixels is illustrated in Figure 4.5.

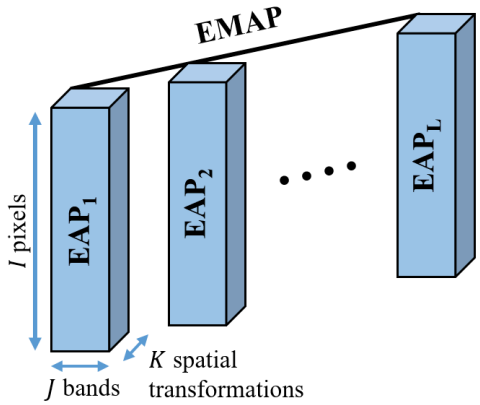


Figure 4.5: An illustration of fourth-order tensor using EMAP after merging the first two modes of pixels

Because of the huge dimensions of our data especially that of the first mode (the high number of pixels), directly applying CPD is exhaustive. One way to do so is through lossless compression, i.e. by compressing the data to the minimal dimensions such that all information is conserved. For example, for a fourth-order tensor of size $I \times J \times K \times L$, the dimensionality

of the mode of pixels can be very big compared to the product of those of the other modes (i.e. $I \gg JKL$). Thanks to HOSVD, such tensor can be compressed to a size of $(JKL) \times J \times K \times L$ supposing that $R_+ \leq JKL$. The compressed tensor is denored by $\boldsymbol{\mathcal{G}}$.

HOSVD is explained in details in Section 2.4.1 having the following expression:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{G}} \underset{1}{\bullet} \boldsymbol{V}_1 \underset{2}{\bullet} \boldsymbol{V}_2 \underset{3}{\bullet} \boldsymbol{V}_3 \underset{4}{\bullet} \boldsymbol{V}_4, \tag{4.6}$$

where $\{\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3, \boldsymbol{V}_4\}$ are orthogonal matrices that can be obtained using SVD of the tensor mode-unfoldings. We note that if there is no need to compress a mode d, then $\boldsymbol{V}_d$ can be set to the identity matrix. The goal is to obtain the compressed tensor $\boldsymbol{\mathcal{G}}$ and use CPD such that:

$$\boldsymbol{\mathcal{G}} = \boldsymbol{\Lambda} \underset{1}{\bullet} \boldsymbol{H}_{1_c} \underset{2}{\bullet} \boldsymbol{H}_{2_c} \underset{3}{\bullet} \boldsymbol{H}_{3_c} \underset{4}{\bullet} \boldsymbol{H}_{4_c}, \tag{4.7}$$

where $\boldsymbol{\Lambda}$ is a diagonal core tensor and $\{\boldsymbol{H}_{1_c}, \boldsymbol{H}_{2_c}, \boldsymbol{H}_{3_c}, \boldsymbol{H}_{4_c}\}$ are the compressed factor matrices. Then, the CPD of $\boldsymbol{\mathcal{T}}$ is expressed as:

$$\begin{aligned} \boldsymbol{\mathcal{T}} &= \boldsymbol{\Lambda} \underset{1}{\bullet} (\boldsymbol{V}_1 \boldsymbol{H}_{1_c}) \underset{2}{\bullet} (\boldsymbol{V}_2 \boldsymbol{H}_{2_c}) \underset{3}{\bullet} (\boldsymbol{V}_3 \boldsymbol{H}_{3_c}) \underset{4}{\bullet} (\boldsymbol{V}_4 \boldsymbol{H}_{4_c}) \\ &= \boldsymbol{\Lambda} \underset{1}{\bullet} \boldsymbol{H}_1 \underset{2}{\bullet} \boldsymbol{H}_2 \underset{3}{\bullet} \boldsymbol{H}_3 \underset{4}{\bullet} \boldsymbol{H}_4 \end{aligned} \tag{4.8}$$

Algorithm 8 shows our implementation to compress any tensor using HOSVD; we note that $\boldsymbol{way}_c$ is a vector containing the modes that are desired to be compressed, and $\boldsymbol{dim}_c$ is a vector of the desired compressed dimensions (i.e. in case there is a desire to truncate the singular vectors of $\boldsymbol{V}_d$).

Algorithm 8 COMPRESS

Require: $\boldsymbol{\mathcal{T}}$, $\boldsymbol{way_c}$, $\boldsymbol{dim_c}$
  for $d \in \boldsymbol{way_c}$ **do**
    Unfold $\boldsymbol{\mathcal{T}}$ into $\boldsymbol{T}_{(d)}$ such that the mode d takes the second way of the matrix;
    Compute the right singular matrix from the SVD of $\boldsymbol{T}_{(d)}$, denoted by $\boldsymbol{V}_d$;
    Truncate the columns of $\boldsymbol{V}_d$ by $\boldsymbol{dim_c}(d)$;
  end for
  for $d \notin \boldsymbol{way_c}$ **do**
    $\boldsymbol{V}_d = \boldsymbol{I}_{\boldsymbol{dim_c(d)}}$; (Identity matrix)
  end for
  $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{T}} \bullet_1 \boldsymbol{V}_1^\mathsf{T} \bullet_2 \ldots \bullet_N \boldsymbol{V}_N^\mathsf{T}$;
  return $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_N$

Computing CPD of a compressed tensor when nonnegative constraints are involved can be tricky to implement, and the issue has been addressed in [27]. Recently, AO-ADMM was introduced in [53] as a powerful algorithm to compute CPD and a flexible one with constraints, which we addressed in Section 2.4.5.4. This being said, in the first part of this section, we talk about our implementation of these constraints in Compressed AO-ADMM. In the second part, we dive into our specific application and talk about the advantages of combining spectral and spatial features using tensor decomposition in the framework of classification.

4.3.1 Compressed AO-ADMM

In our case, we are mainly interested in compressing the first mode, and the compressed tensor $\boldsymbol{\mathcal{G}}$ is the one that is used for decomposition with CPD. However, $\boldsymbol{\mathcal{G}}$ can contain negative entries, while $\boldsymbol{\mathcal{T}}$ is nonnegative and its factor matrices are constrained to be so. As such, only $\boldsymbol{H}_1$ is compressed, so assuming that $\boldsymbol{H}_1 = \boldsymbol{V}_1\boldsymbol{H}_{1_c} \succeq 0$ (where $\boldsymbol{H}_{1_c}$ is the compressed version of $\boldsymbol{H}_1$), the optimization problem (for Nth-order) can be reformulated such that:

$$\begin{aligned} &\underset{\boldsymbol{H}_{1_c},\ldots,\boldsymbol{H}_N}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\mathcal{G}} - (\boldsymbol{H}_{1_c},\ldots,\boldsymbol{H}_N).\boldsymbol{\Lambda}\|^2 \\ &\text{s.t. } \boldsymbol{V}_1\boldsymbol{H}_{1_c} \succeq 0, \boldsymbol{H}_2 \succeq 0, \ldots, \boldsymbol{H}_N \succeq 0 \end{aligned} \tag{4.9}$$

In order to cope with the constraints of compression and nonnegativity at once, we present a solution that is inspired by [27], with AO-ADMM being the adopted algorithm for its efficiency and flexibility with constraints [53]. Using ADMM, it is possible to solve the sub-problem of each factor matrix with the corresponding constraints and alternate accordingly. Hence, solving (4.9) boils down to the following updates, defined for $d = \{2,\ldots,N\}$:

$$\left.\begin{aligned} \tilde{\boldsymbol{H}}_{1_c} &\leftarrow (\boldsymbol{W}^\mathsf{T}\boldsymbol{W} + \rho\boldsymbol{I})^{-1}(\boldsymbol{W}^\mathsf{T}\boldsymbol{G}_{(1)} + \rho(\boldsymbol{U} + \boldsymbol{H}_{1_c})^\mathsf{T}) \\ \boldsymbol{H}_{1_c} &\leftarrow \boldsymbol{V}_1^\mathsf{T}\max(0, \boldsymbol{V}_1(\tilde{\boldsymbol{H}}_{1_c}^\mathsf{T} - \boldsymbol{U})) \\ \boldsymbol{U} &\leftarrow \boldsymbol{U} + \boldsymbol{H}_{1_c} - \tilde{\boldsymbol{H}}_{1_c}^\mathsf{T} \end{aligned}\right\} \tag{4.10}$$

$$\left.\begin{aligned} \tilde{\boldsymbol{H}}_{d} &\leftarrow (\boldsymbol{W}^\mathsf{T}\boldsymbol{W} + \rho\boldsymbol{I})^{-1}(\boldsymbol{W}^\mathsf{T}\boldsymbol{G}_{(d)} + \rho(\boldsymbol{U} + \boldsymbol{H}_{d})^\mathsf{T}) \\ \boldsymbol{H}_{d} &\leftarrow \max(0, \tilde{\boldsymbol{H}}_{d}^\mathsf{T} - \boldsymbol{U}) \\ \boldsymbol{U} &\leftarrow \boldsymbol{U} + \boldsymbol{H}_{d} - \tilde{\boldsymbol{H}}_{d}^\mathsf{T} \end{aligned}\right\} \tag{4.11}$$

We show through Algorithms 9, 10, and 11 how we implement the AO-ADMM method. In Algorithm 10, $\boldsymbol{T}$ is a matrix updated by $\boldsymbol{T}_{(d)}$, $\boldsymbol{W}$ is the Khatri-Rao product defined in Algorithm 9, k is the input rank of the decomposition, μ is a regularization paramter for which we refer to [53] for its updates, ϵ is the tolerance for r and s which we refer to [53] for their updates, *constraint* is a string defining the type of proximity update, i_{max} is the number of inner ADMM iterations (0 if no limit), and $\boldsymbol{V}_d$ is the compression matrix. For Section 4.3.1.1, we note that the dimensions of $\boldsymbol{T}_{(d)}$ are $\prod_{j\neq d} I_j \times I_d$, those of $\boldsymbol{W}$ are $\prod_{j\neq d} I_j \times R$, and those of $\tilde{\boldsymbol{H}}_d$, $\boldsymbol{H}_d^\mathsf{T}$ and $\boldsymbol{U}^\mathsf{T}$ are $R \times I_d$.

4.3.1.1 Computational complexity

Concerning the computational complexity of AO-ADMM for CPD, we refer to paper [53] for the detailed explanations. Here, we compare the case of compressed AO-ADMM to that of TPCA. Let us consider a third-order tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I_1\times I_2\times I_3}$ of rank R, and we consider the complexity as per ADMM iterations (Algorithm 10). Also, it is important to note that the matrices $\boldsymbol{W}$ and $\boldsymbol{T}$ are independent from the inner-ADMM updates, so they can be used only

once to compute the Cholesky decomposition and the product $\boldsymbol{W}^\top\boldsymbol{T}$, whose values can be cached before the $\tilde{\boldsymbol{H}}$ update allowing to save a lot of repetitive computations. Now, we split the cases into four:

1. Unconstrained CPD: In this case, the complexity of the algorithm is dominated only by the updates of $\tilde{\boldsymbol{H}}$. Hence, the complexity is $\boldsymbol{\mathcal{O}}(I_d R^2)$.

2. Nonnegative CPD: Nonnegative constraints on the factor matrices require only element-wise projection, i.e. a complexity of $\boldsymbol{\mathcal{O}}(I_d R)$, which is negligible compared to $\boldsymbol{\mathcal{O}}(I_d R^2)$. Hence, the same complexity of unconstrained CPD still dominates.

3. Unconstrained CPD of the compressed tensor (compression along the mode of pixels): This case requires one step after the whole AO-ADMM framework to decompress the first factor matrix back into $\boldsymbol{H}_1$, which is the final product before the classification phase. However, the ADMM updates by themselves remain unchanged with a gain on the complexity $\boldsymbol{\mathcal{O}}(I_1 R^2)$ since I_1 here is compressed to a much lower value, i.e. $I_2 I_3$. The complexity of the ADMM updates generally remains dominated by $\boldsymbol{\mathcal{O}}(I_2 I_3 R^2)$ per iteration, and that of the decompression step is $\boldsymbol{\mathcal{O}}(I_1 I_2 I_3 R)$.

4. Nonnegative CPD of the compressed tensor (compression along the mode of pixels): This case affects only the ADMM updates of the first factor matrix, specifically at the level of $\boldsymbol{H}_{1_c}$ in equation (4.10) due to the decompression and compression steps. The complexity increases by $\boldsymbol{\mathcal{O}}(I_1 I_2 I_3 R)$ per iteration for the first factor matrix $\boldsymbol{H}_{1_c}$, but remains unchanged for the others (i.e. $\boldsymbol{\mathcal{O}}(I_d R^2)$, $d = 2, 3$).

In the case of TPCA, a PCA is carried out for every matrix $Y_{(d)}$ that corresponds to a feature mode in order to finally perform the tensor product that would reduce the dimensionality of the feature modes separately. This means that for the same third-order tensor considered earlier, two PCA steps are carried out. Each PCA holds as much complexity to it as that of an SVD, that is $\boldsymbol{\mathcal{O}}(mn^2)$ for a matrix of dimensions $m \times n$ where $m \geq n$. Considering in this case that the PCA is carried out for the second and third matrix-unfoldings, i.e. $Y_{(2)}$ and $Y_{(3)}$, then the complexities are $\boldsymbol{\mathcal{O}}(I_1 I_2 I_3 I_2)$ and $\boldsymbol{\mathcal{O}}(I_1 I_2 I_3 I_3)$ respectively. After that, in order to obtain the tensor with reduced feature dimensions, say $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{I_1 \times R_2 \times R_3}$, through tensor product, the complexity is as much as $\boldsymbol{\mathcal{O}}(I_1 R_2 R_3 I_2 I_3)$.

For the sake of comparison, this means that for each additional feature mode of dimension I_N in the tensor, the complexity of the CPD in the first three cases will only add by $\boldsymbol{\mathcal{O}}(I_N R^2)$ (one additional matrix update). In the fourth case, the complexity of the decompression and compression step increases to $\boldsymbol{\mathcal{O}}(I_1 \dots I_N R)$ (multiplies once by the new dimension). In the case of TPCA, it increases by $\boldsymbol{\mathcal{O}}(I_1 \dots I_N I_d)$ for every $d = \{1, \dots, N\}$, with an additional load at the tensor product phase, i.e. $\boldsymbol{\mathcal{O}}(I_1 R_2 \dots R_N I_2 \dots I_N)$. In general, R is much smaller than, or at most comparable to, $\prod_{d=2}^{N} R_d$. Finally, the computational complexity is better in the case of CPD for most cases, but the choice of the parameters can affect this conclusion.

Algorithm 9 The Alternating Optimization part of solving (4.9), it alternates between the ADMMs of the factor matrices

Require: $\boldsymbol{\mathcal{T}}$, $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N$, $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N$
  $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N$ are initialized; $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N$ are initialized to zero matrices;
  for $d = 1, \ldots, N$ **do**
    Store the different unfoldings $\boldsymbol{T}_{(d)}$;
  end for
  repeat
    for $d = 1, \ldots, N$ **do**
      $\boldsymbol{W} = \odot_{j \neq d} \boldsymbol{H}_j$;
      update $\boldsymbol{H}_d$ and $\boldsymbol{U}_d$ using Algorithm 10; (a single ADMM, i.e. (4.10) and (4.11))
    end for
    update μ if necessary; (refer to [53] for the update of the regularization parameter μ)
  until some termination criterion is reached (number of iterations)
  Normalize the columns of the factor matrices and store the weights in $\boldsymbol{\Lambda}$;
  return $\boldsymbol{H}_1, \ldots, \boldsymbol{H}_N$, $\boldsymbol{\Lambda}$

4.3.2 Spectral and Spatial Features Using CP Decomposition

Now, in our particular case of HSI tensors, and based on what was discussed in section 4.2, we recall that the first mode spans the pixels, the second mode spans the spectral bands, the third mode spans morphological transformations, and the possible fourth mode spans different kinds of morphological concepts. For simplicity, we keep the following explanation short to third-order tensors, higher-order tensors then follow analogously.

We show an illustration of some relationships between a third-order tensor and the components of its CPD in Figure 4.6. Here, we expect that $\boldsymbol{H}_1$ represents the mode of pixels, $\boldsymbol{H}_2$ represents that of spectral bands, and $\boldsymbol{H}_3$ represents the corresponding morphological diversity. Accordingly, each row in $\boldsymbol{H}_1$ (say at index p as pointed in the figure) describes the synthesis or composition of a high-order pixel of $\boldsymbol{\mathcal{T}}$ having the same index in the first mode (i.e. by fixing the index of the first mode at p such as the horizontal green slab in the figure), which is reflected element-wise as follows:

$$\begin{gathered} t_{\mathbf{p},i_2,i_3} = \sum_{r=1}^{R_+} h^{(1)}_{\mathbf{p},r} \cdot h^{(2)}_{i_2,r} \cdot h^{(3)}_{i_3,r} \cdot \lambda_r \\ \forall\, i_2 \in \{1, \ldots, I_2\}, \;\; \forall\, i_3 \in \{1, \ldots, I_3\} \end{gathered} \tag{4.12}$$

Additionally, the coefficients found in one row in $\boldsymbol{H}_1$ (across the columns) are related to their counterpart-columns in $\boldsymbol{H}_2$ and $\boldsymbol{H}_3$, both of which represent feature information (Figure 4.6 highlights in orange similarly indexed columns from each of the factor matrices along with the corresponding coefficient in $\boldsymbol{\Lambda}$). In other words, these coefficients in $\boldsymbol{H}_1$ describe the spectral and morphological information that spread out in $\boldsymbol{H}_2$ and $\boldsymbol{H}_3$ respectively. Consequently, $\boldsymbol{H}_1$ can be seen as a matrix of samples and features where spectral and spatial features are factorized.

Algorithm 10 ADMM update of $\boldsymbol{H}_d$ and $\boldsymbol{U}_d$, i.e., (4.10) and (4.11) given a mode d

Require: $\boldsymbol{T}$, $\boldsymbol{W}$, $\boldsymbol{H}$, $\boldsymbol{U}$, k, μ, ϵ, *constraint*, i_{max}, $\boldsymbol{V}$
  $\boldsymbol{H}$ and $\boldsymbol{U}$ are already initialized;
  $\boldsymbol{G} = \boldsymbol{W}^\mathsf{T}\boldsymbol{W}$;
  $\rho = \text{trace}(\boldsymbol{G})/k$;
  Calculate $\boldsymbol{L}$ from Cholesky decomposition such that
  $\boldsymbol{G} + (\rho + \mu)\boldsymbol{I_k} = \boldsymbol{L}\boldsymbol{L}^\mathsf{T}$;
  $\boldsymbol{F} = \boldsymbol{W}^\mathsf{T}\boldsymbol{T}$;
  $\boldsymbol{H_f} = \boldsymbol{H}$;
  repeat
    $\tilde{\boldsymbol{H}} \leftarrow (\boldsymbol{L}^\mathsf{T})^{-1}\boldsymbol{L}^{-1}(\boldsymbol{F} + \rho(\boldsymbol{H} + \boldsymbol{U})^\mathsf{T} + \mu\boldsymbol{H_f}^\mathsf{T})$; (See [53] for μ and $\boldsymbol{H_f}$)
    $\boldsymbol{H} \leftarrow$ proximity(*constraint*,$\tilde{\boldsymbol{H}}^\mathsf{T}$,$\boldsymbol{U}$,$\boldsymbol{V}$); (refer to Algorithm 11)
    $\boldsymbol{U} \leftarrow \boldsymbol{U} + \boldsymbol{H} - \tilde{\boldsymbol{H}}^\mathsf{T}$;
    Update r and s; (refer to [53] for the updates of r and s)
  until ($r < \epsilon$ and (s is undefined or $s < \epsilon$)) or ($imax > 0$ and $i \geq imax$)
  return $\boldsymbol{H}$ and $\boldsymbol{U}$

Algorithm 11 Proximity Update of $\boldsymbol{H}$ in ADMM

Require: *constraint*, $\boldsymbol{H}_t$, $\boldsymbol{U}$, $\boldsymbol{V}$
  switch (*constraint*)
  case Nonnegativity**:**
    $\boldsymbol{H} \leftarrow \boldsymbol{H}_t - \boldsymbol{U}$;
    $\boldsymbol{H} \leftarrow \max(0, \boldsymbol{H})$;
  case Compression and Nonnegativity**:**
    $\boldsymbol{H}_u \leftarrow \boldsymbol{V}(\boldsymbol{H}_t - \boldsymbol{U})$;
    $\boldsymbol{H} \leftarrow \boldsymbol{V}^\mathsf{T} \max(0, \boldsymbol{H}_u)$;
  end switch
  return $\boldsymbol{H}$

Thanks to the nonnegative constraints in the decomposition, the columns of the factor matrices can hold physical interpretation. For instance, the columns of $\boldsymbol{H}_2$ can be seen as spectral signatures, and those of $\boldsymbol{H}_1$, when folded (reshaped into matrices), can be seen as corresponding grayscale images. This can also be useful for Hyperspectral Unmixing, which is discussed in Chapter 5 because it is out of the scope of this chapter.

Briefly speaking, CP decomposition directly provides a simple and low dimensional representation of the data. First, the complexity of the tensor is reduced to simple matrices, each linked to one of its modes, with an intuitive approach to account to the high-order relationships in the data and sometimes with barely any loss of information. Second, the high-order feature aspect of the pixels in $\boldsymbol{\mathcal{T}}$ boils down to row vectors of R_+ elements in $\boldsymbol{H}_1$. Third, there are often redundancies in the tensor, which makes R_+ relatively small and corresponds to the low dimensionality of the feature space. Finally, since classification is our main concern, our target is to classify *the matrix* representing the pixel mode, $\boldsymbol{H}_1$. We think of the rows as samples and the columns as features in the classification. For that, we conduct some

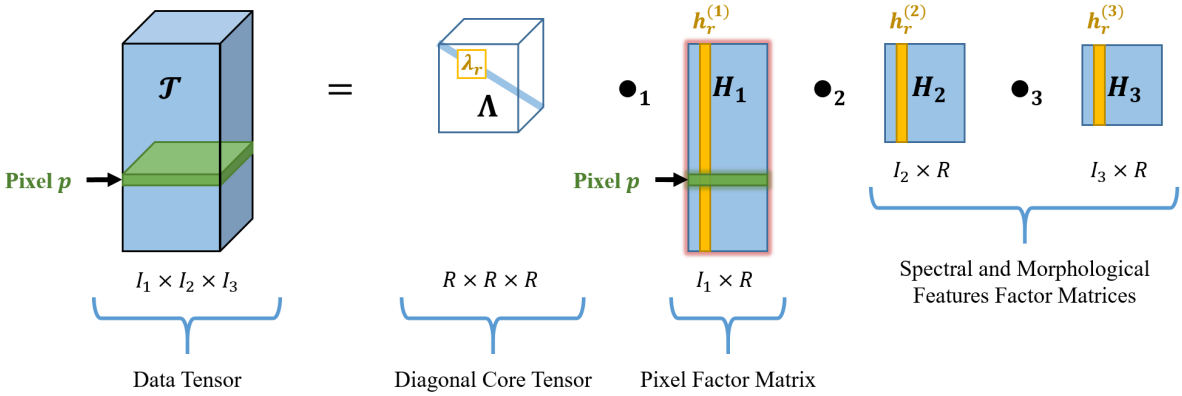


Figure 4.6: An illustration of some relationships between the tensor and its CPD. "Pixel p" refers to a pixel at index p. $\boldsymbol{h}_r^{(1)}$, $\boldsymbol{h}_r^{(2)}$, and $\boldsymbol{h}_r^{(3)}$, are the r-th columns of matrices $\boldsymbol{H}_1$, $\boldsymbol{H}_2$, and $\boldsymbol{H}_3$ respectively. λ_r is the r-th diagonal element of $\boldsymbol{\Lambda}$.

experiments that are to be shown in the following section.

4.4 Experiments and Results

In this section, we talk about the experiments conducted in this work. First, we give a description of the data sets that we use. Then we present the experimental set-ups and their corresponding results for further discussion. In order to demonstrate our method, we use two real HSI datasets as shown in section 4.4.1.

In the results sections, we demonstrate the results on the Pavia University HSI first, then we show those of the Data Fusion Contest HSI (DFC). For each HSI, we show three different kinds of morphological applications, where the pixels are vectorized afterwards:

- EMP: Based on morphological operators. The decomposed tensor is third-order. The spatial way is the stacking of the transformations as they are.
- NAMD: Based on morphological operators. The decomposed tensor is third-order. The spatial way is a form of a successive differential stacking of the transformations.
- EMAP: Based on attribute operators. The decomposed tensor is fourth-order. One spatial way corresponds to attribute indexing (the kind of attribute), and the other one is the stacking of the transformations as they are.

With EMP and NAMD, our approach is only compared to that of TPCA as the tensor-based state of the art technique in order to focus on an analysis regarding some parameters mentioned below, while with EMAP, our approach is compared to that of PCA and NMF of the original image, classical EMAP of all the features concatenated along one mode, PCA of classical EMAP, and TPCA and CPD of tensor-based EMAP. In each of the CPD experiment,

among 20 random initializations of the factor matrices, the result with the minimum root mean squared error (RMSE) is chosen.

Moreover, we make some remarks on different aspects of the used decomposition. As a start, we notice that there are two main variables that can influence the decomposed data; the number of AO-ADMM iterations and the rank of the decomposition, which are usually provided as inputs to the CPD. For instance, the higher the number of iterations or the value of the rank is, the less the reconstruction error, so we expect the decomposed data to better represent the original tensor up to overfitting. Another aspect would be the constraints. Since we use nonnegativity constraints on the original factor matrices while carrying the CPD on the compressed versions, the decomposition can take a lot of time compared to its unconstrained counterpart, so we show some experiments using unconstrained CPD and give some remarks. On a side note, NCPD is also useful for high-order Hyperspectral Unmixing, which shows how it is possible to identify the patterns of the feature factor matrices corresponding to the columns of the pixel factor matrix, and this point is discussed in Chapter 5.

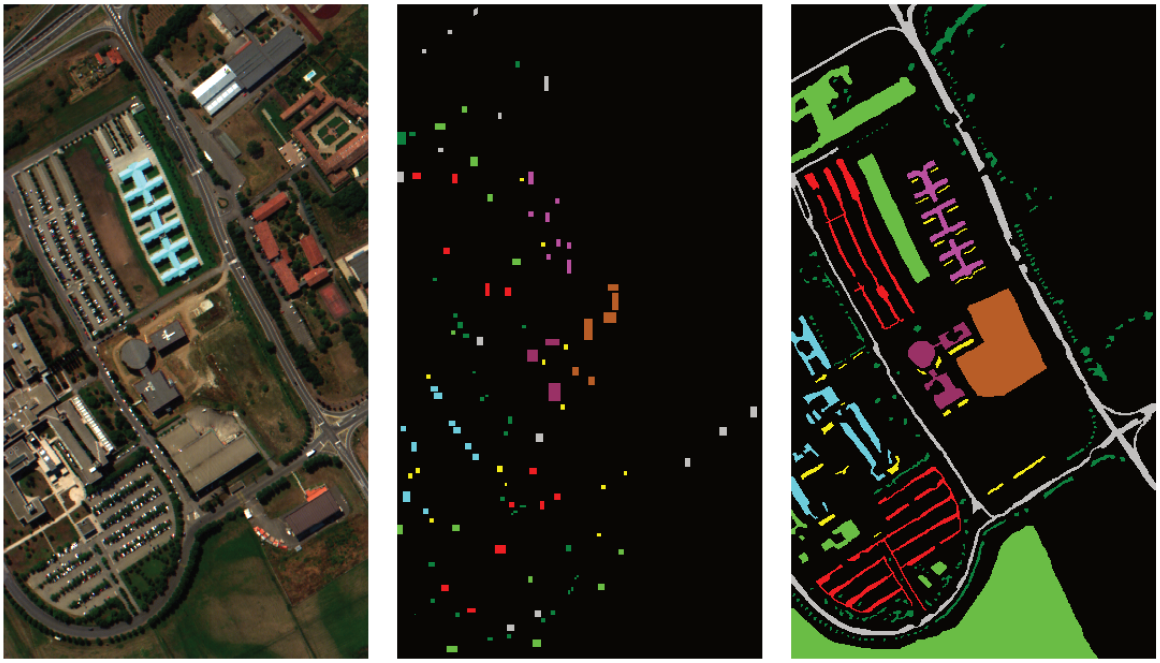
4.4.1 Description of Data Sets

4.4.1.1 Pavia University

The first HSI, known as Pavia University, was taken over the University of Pavia and acquired by the ROSIS sensor. The image has a spatial size of 610×340 pixels with a geometric resolution of 1.3 meters, and consists of 103 spectral bands. The groundtruth image is included besides the data set and it consists of nine classes: trees, asphalt, bitumen, gravel, metal sheets, shadows, self-blocking bricks, meadows, and bare soil. Additionally, 40002 pixels are available as test set, and 3921 pixels are available as training set. Figure 4.7 shows the HSI in false colors (by choosing the bands 58, 34, and 17 as Red, Green and Blue components) as well as the training set and ground-truth pixels.

4.4.1.2 Data Fusion Contest (DFC) image for IEEE GRSS 2013

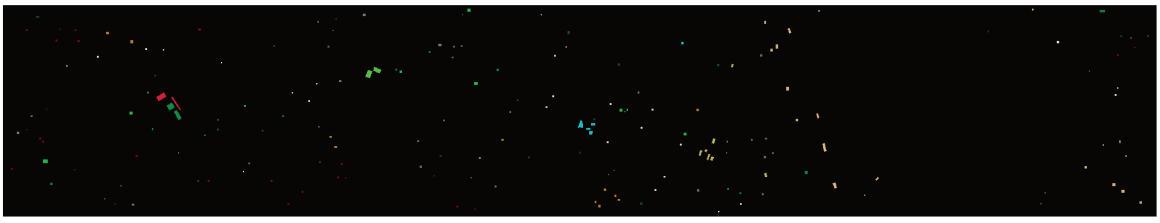
The second HSI was acquired over the University of Houston campus and the neighboring urban area by the ITRES-CASI 1500 sensor. The image has a spatial size of 349×1905 pixels with a geometric resolution of 2.5m, and consists of 144 spectral bands. The groundtruth image is included in the data set and it consists of 15 classes: grass healthy, grass stressed, grass synthetic, tree, soil, water, residential, commercial, road, highway, railway, parking lot 1, parking lot 2, tennis court, and running track. Additionally, 12197 pixels are available as test set, and 2832 pixels are available as training set. Figure 4.8 shows the HSI in false colors (by choosing the bands 65, 40, and 22) as well as the training set and ground-truth pixels.



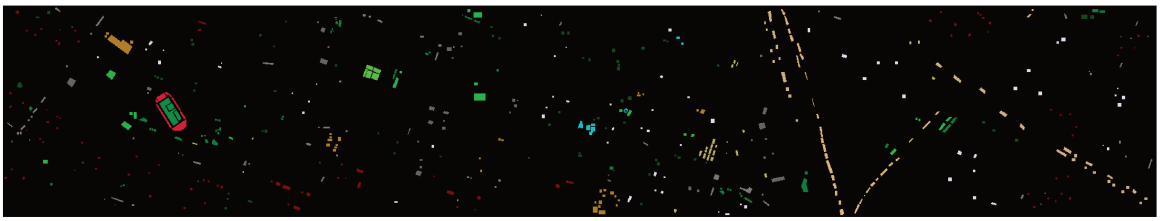
(a) PU. False colors (b) PU. Training set (c) PU. Ground-truth

Figure 4.7: Pavia University HSI in false colors with the training set and ground-truth

(a) DFC. False colors



(b) DFC. Training set



(c) DFC. Ground-truth

Figure 4.8: DFC HSI in false colors with the training set and ground-truth

4.4.1.3 Some Remarks Regarding the Experimental Parameters

In the following, to avoid repetition, we describe some common parameters used in the morphological methods.

Concerning the choice of the parameters of the morphological and attribute transformations, they where chosen such that enough distinctive variations or features can be observed in the profiles along the third mode (in order to reduce redundat spatial features).

As for the CPD, in some of the following experiments, we would like to see the effect of the number of iterations and the choice of the rank on the results. The number of iterations is generally set to 50, which is chosen by looking at the plot of the reconstruction error of (4.9) with respect to the number of iterations. However, some experiments in Sections 4.4.2 were carried out to compare between the cases of 30 and 50 iterations (starting with the same initial conditions). As for the value of the rank, first, we note that it sets the value of the reduced dimension of the feature space. Second, we compare the results between different values where we start with a value that is very close to the number of predefined classes in the groundtruth image (say, 10 or 15), then we test higher values up to overfitting. Note that the expression $\text{CPD}_{(i,R)}$ refers to a CPD carried out with i iterations and rank R.

TPCA is one of the methods that we compare our results to. In this regard, the number of principal components of the modes are decided based partly on the elbow rule of the graphs of the singular values obtained from the SVD of the corresponding mode-unfolding of the tensor, and partly according to how much the reduced mode dimensions can explain the original data. Note that the expression $\text{TPCA}_{(pc_3,pc_4)}$ refers to a TPCA where the dimensions of the spectral and morphological modes are reduced to pc_3 and pc_4 PCs respectively before vectorizing the pixels, and $\text{TPCA}_{(pc_3,pc_4,pc_5)}$ follows suite (i.e., in the case of EMAP).

In the classification phase, Support Vector Machine (SVM) with a Gaussian kernel was used. The hyper-parameters of SVM were optimized using 5-fold cross-validation as mentioned in the guide of [52]. The training and testing sets that are available with the two data-sets were used to train and test the classifier.

4.4.2 Experiments with Pavia University

4.4.2.1 EMP

First, we start by showing the results of the EMP method. In this part, we use Opening and Closing by reconstruction as the operators ($\bar{\Phi} = \gamma^{\infty}_{\mathtt{SE}}, \underline{\Phi} = \phi^{\infty}_{\mathtt{SE}}$). In the following, we fix the parameters of the transformations to $m = 6$ disk-shaped structuring elements of different sizes, in pixels: $\{1, 6, 11, 16, 21, 26\}$. This means that the fourth mode of the tensor is of dimension $K = 13$ with the arrangement shown in (4.2). The first two modes are then merged to give tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{207400\times103\times13}$.

Before applying CP decomposition, since the data is huge, we attempt to pre-process the tensor by compressing its pixel mode as explained in Section 4.3. After the compression step, we obtain a tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{1339\times 103\times 13}$ (i.e., the dimension of the first mode is reduced to $103 * 13 = 1339$), which becomes the input to be directly decomposed through CP decomposition.

Now, we compute the CP decomposition with nonnegativity constraints as mentioned in the previous sections. We also use different values of the rank in order to see the variations in the results such that $R = \{10, 20, 30, 40\}$. The factor matrices are initialized randomly as the absolute value of the i.i.d standard Gaussian distribution, and only the first factor matrix is compressed afterwards.

For TPCA (applied before vectorizing the pixels), four to six principal components were chosen for the third-mode tensor unfolding explaining the data by 98.97%, 99.28%, and 99.47% respectively, and four to six components were chosen for the fourth-mode tensor unfolding explaining the data by 98.95%, 99.37%, and 99.57% respectively. As a result, this maps to nine different classification results noting that the total number of features is the product of the number of the principal components chosen from the latter two modes.

Discussion. In Table 4.1, we show some tests with their reconstruction errors. Indeed, we notice that the more iterations or the higher the rank, the less the reconstruction error, and generally, the better the classification accuracy. Increasing the number of iterations allows AO-ADMM to converge further, but at some point there will be only slight changes. Increasing the rank allows for more degrees of freedom for the data to be spread in the decomposition, but first we need the rank to be relatively small, second we notice that eventually there will be some kind of a limit for how much the reconstruction error and the classification accuracy can improve, and third if the rank is high then noisy structures may start to appear in the decomposition.

Method	No. of Features (R)	Rec. Error %	OA %	AA %
EMP + $\text{TPCA}_{(5,5)}$	25	-	93.20	91.16
EMP + $\text{CPD}_{(30,10)}$	10	6.19	91.82	91.18
EMP + $\text{CPD}_{(50,10)}$	10	6.16	<u>93.25</u>	<u>91.58</u>
EMP + $\text{CPD}_{(30,20)}$	20	4.20	<u>93.24</u>	<u>92.33</u>
EMP + $\text{CPD}_{(50,20)}$	20	4.06	<u>93.99</u>	<u>92.36</u>
EMP + $\text{CPD}_{(30,30)}$	30	3.18	<u>94.34</u>	<u>93.54</u>
EMP + $\text{CPD}_{(50,30)}$	30	3.05	<u>95.63</u>	<u>95.41</u>
EMP + $\text{CPD}_{(30,40)}$	40	2.68	<u>97.73</u>	<u>97.71</u>
EMP + $\text{CPD}_{(50,40)}$	40	2.59	<u>97.68</u>	<u>97.71</u>

Table 4.1: Pavia University. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMP set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The CPD cases are grouped by the same value of the rank. The cases when CPD does better are underlined.

In addition to the influence of the number of iterations and the rank, NCPD was able to bring better accuracy results starting at only 10 features compared to 25 features for TPCA. Figures 4.10a-4.10e show some classification maps for the EMP setup, with an indication on the methods used and their parameters.

4.4.2.2 NAMD

Now we show some of the results using the NAMD method. Similarly to EMP, Opening and Closing were chosen as the morphological operators with the same type of the structuring element and same size values. At the end we obtain a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{207400\times103\times13}$, which is then compressed into $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{1339\times103\times13}$.

We use the same parameters of the NCPD as those in the EMP setup. For TPCA, three to five principal components were chosen for the third-mode tensor unfolding explaining the data by 99.67%, 99.75%, and 99.82% respectively, and four to six components were chosen for the fourth-mode tensor unfolding explaining the data by 97.72%, 98.65%, and 99.47% respectively.

Discussion. We show some of the results in Table 4.2. We notice the same pattern of reconstruction error and classification accuracy results as those found in Table 4.1 for increasing number of iterations and values of the rank though the reconstruction errors are higher in Table 4.2. NCPD was able to bring better accuracy results starting at 30 features compared to 30 features for TPCA. Figures 4.10f-4.10j show some classification maps for the NAMD setup, with an indication on the methods used and their parameters.

Method	No. of Features (R)	Rec. Error %	OA %	AA %
NAMD + $\text{TPCA}_{(5,6)}$	30	-	92.23	90.94
NAMD + $\text{CPD}_{(30,10)}$	10	18.02	87.21	88.46
NAMD + $\text{CPD}_{(50,10)}$	10	17.98	87.77	88.28
NAMD + $\text{CPD}_{(30,20)}$	20	11.21	91.35	88.78
NAMD + $\text{CPD}_{(50,20)}$	20	11.10	92.16	89.77
NAMD + $\text{CPD}_{(30,30)}$	30	8.76	<u>94.52</u>	<u>91.77</u>
NAMD + $\text{CPD}_{(50,30)}$	30	8.71	<u>94.59</u>	<u>91.72</u>
NAMD + $\text{CPD}_{(30,40)}$	40	7.24	<u>96.96</u>	<u>96.44</u>
NAMD + $\text{CPD}_{(50,40)}$	40	7.23	<u>96.91</u>	<u>96.49</u>

Table 4.2: Pavia University. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the NAMD set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The CPD cases are grouped by the same value of the rank. The cases when CPD does better are underlined.

4.4.2.3 EMAP

In this setting, we start by showing the four types of attributes and their corresponding threshold values that we used. Some attribute thresholds depend on the size of objects in the image, and others depend on the range of values in the pixels, so they were chosen based on observed changes between transformations:

- Area of the regions; $\lambda_a = \{100, 500, 1000, 5000\}$.
- Diagonal of the bounding box; $\lambda_d = \{10, 25, 50, 100\}$.
- Standard deviation; $\lambda_s = \{20, 30, 40, 50\}$.
- Moment of inertia; $\lambda_i = \{0.2, 0.3, 0.4, 0.5\}$.

First, we refer to [31] for the definitions of the attributes. For each attribute, we fix four different thresholds, corresponding to eight thinnings and thickenings, which means that the fourth mode is of dimension $K = 9$ (including the original image). After we consider the different attributes together in one data block, the fifth mode becomes of dimension $L = 4$. After merging the first two modes, we obtain a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{207400\times103\times9\times4}$.

We notice that, practically, compressing or decomposing $\boldsymbol{\mathcal{T}}$ can be computationally very demanding, so we tend to reduce the spectral dimension in the original HSI, $\boldsymbol{\mathcal{I}}$. In order to do that and conserve both the nonnegativity and the information in $\boldsymbol{\mathcal{I}}$, we use Nonnegative Matrix Factorization (NMF) computed through AO-ADMM on its first matrix unfolding, call it $\boldsymbol{I}$, where one mode represents pixels arranged in lexicographic order, and the other represents spectral bands. NMF decomposes $\boldsymbol{I}$ into two other matrices with nonnegative entries, e.g. $\boldsymbol{I} = \boldsymbol{W}\boldsymbol{H}^{\mathsf{T}}$, where $\boldsymbol{W}$ represents pixel information, and the number of columns in $\boldsymbol{W}$ and $\boldsymbol{H}$ is defined by the rank of the NMF, call it R_{NMF}, which is usually relatively small. $\boldsymbol{W}$ is then chosen and rearranged into a data cube, $\boldsymbol{\mathcal{W}}$, with reduced spectral dimension, to be used in order to form $\boldsymbol{\mathcal{T}}$. Suppose that we note by R_{NMF} as the size of the reduced spectra, then in the case of Pavia HSI, $\boldsymbol{\mathcal{T}}$ is of dimensions $207400 \times R_{NMF} \times 9 \times 4$. Now, following what we do with EMP and NAMD, $\boldsymbol{\mathcal{T}}$ is decomposed using compressed NCPD. The steps are visualized in a flowchart as seen in Figure 4.9.

In a similar manner, in the case of TPCA, PCA of $\boldsymbol{\mathcal{I}}$ was computed as means of reducing the dimension of the spectral mode before forming the EMAP on the principal components. After that, $\boldsymbol{\mathcal{T}}$ is formed. The PCA and EMAP part was done in [31].

For NMF, we choose a value 40 of the rank which corresponds to a reconstruction error of 0.65% compared to the original HSI. For PCA, we choose four principal components which explains the data by 99.16%. For CPD, we choose one value of the rank, which is $R = 40$. For TPCA, four principal components were chosen for the third-mode tensor unfolding explaining the HSI by 100%, four components were chosen for the fourth-mode unfolding explaining the data by 99.28%, and three components were chosen for the fifth-mode unfolding explaining the data by 99.98%.

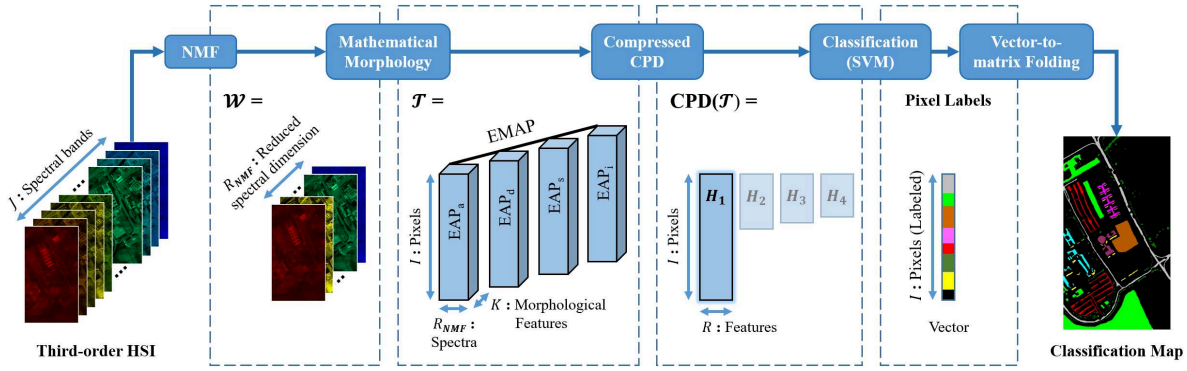


Figure 4.9: A flowchart of the EMAP procedure which includes a preprocessing step to reduce the number of spectral bands.

Discussion. Table 4.3 shows some results related to EMAP. It is important to note that in some cases, the data was rescaled to fit the threshold values of the attribute profiles such that we see enough distinctive features between the images of the attribute profiles. With a lower number of 40 features compared to 48 for TPCA, NCPD showed better accuracy results. This shows that the method can still bring good results when going from an order-4 tensor to a matrix using NCPD. Figures 4.10k and 4.10m show some corresponding classification maps.

Method	No. of Features (R)	Rec. Error %	OA %	AA %
$\text{PCA}_{(4)}$	4	-	68.30	80.28
$\text{NMF}_{(40)}$	40	0.65	77.01	87.65
EMAP	144	-	91.59	92.92
EMAP + $\text{PCA}_{(15)}$	15	-	94.46	94.12
EMAP + $\text{TPCA}_{(4,4,3)}$	48	-	95.44	94.86
EMAP + $\text{CPD}_{(50,40)}$	40	2.03	<u>97.06</u>	<u>98.60</u>

Table 4.3: Pavia University. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMAP set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The cases when CPD does better are underlined.

Finally, Table 4.4 shows per-class results of Overall Accuracy and Average Accuracy of selected examples from Tables 4.1, 4.2 and 4.3 in a side-to-side comparison including the three morphological set-ups (EMP, AMD, and EMAP) and two decomposition methods (TPCA and CPD). From TPCA to CPD, we notice that the accuracy of some classes improved or worsened only slightly, while others such as Gravel, Bare Soil and Shadows improved significantly.

Method	EMP %		NAMD %		EMAP %	
	$\text{TPCA}_{(5,5)}$	$\text{CPD}_{(50,40)}$	$\text{TPCA}_{(5,6)}$	$\text{CPD}_{(50,40)}$	$\text{TPCA}_{(4,4,3)}$	$\text{CPD}_{(50,40)}$
Asphalt	94.21	98.00	96.58	98.63	94.59	98.36
Meadow	94.77	97.11	93.05	97.47	97.60	94.88
Gravel	58.4	94.32	59.55	89.58	77.85	98.84
Tree	99.17	98.14	99.17	98.83	98.04	97.35
Metal Sheet	99.46	99.55	99.73	99.64	99.28	99.46
Bare Soil	89.78	99.71	84.44	92.34	88.01	99.80
Bitumen	99.08	99.28	99.49	99.59	99.79	99.69
Brick	98.84	98.42	98.15	98.12	99.22	99.25
Shadow	86.79	94.84	88.30	94.21	99.37	99.74
OA	93.20	97.68	92.23	96.91	95.44	97.06
AA	91.16	97.71	90.94	96.49	94.86	98.6

Table 4.4: Pavia University. Some per-class accuracy records including those of Overall and Average Accuracies for the EMP, NAMD, and EMAP set-ups.

4.4.3 Experiments with DFC Image

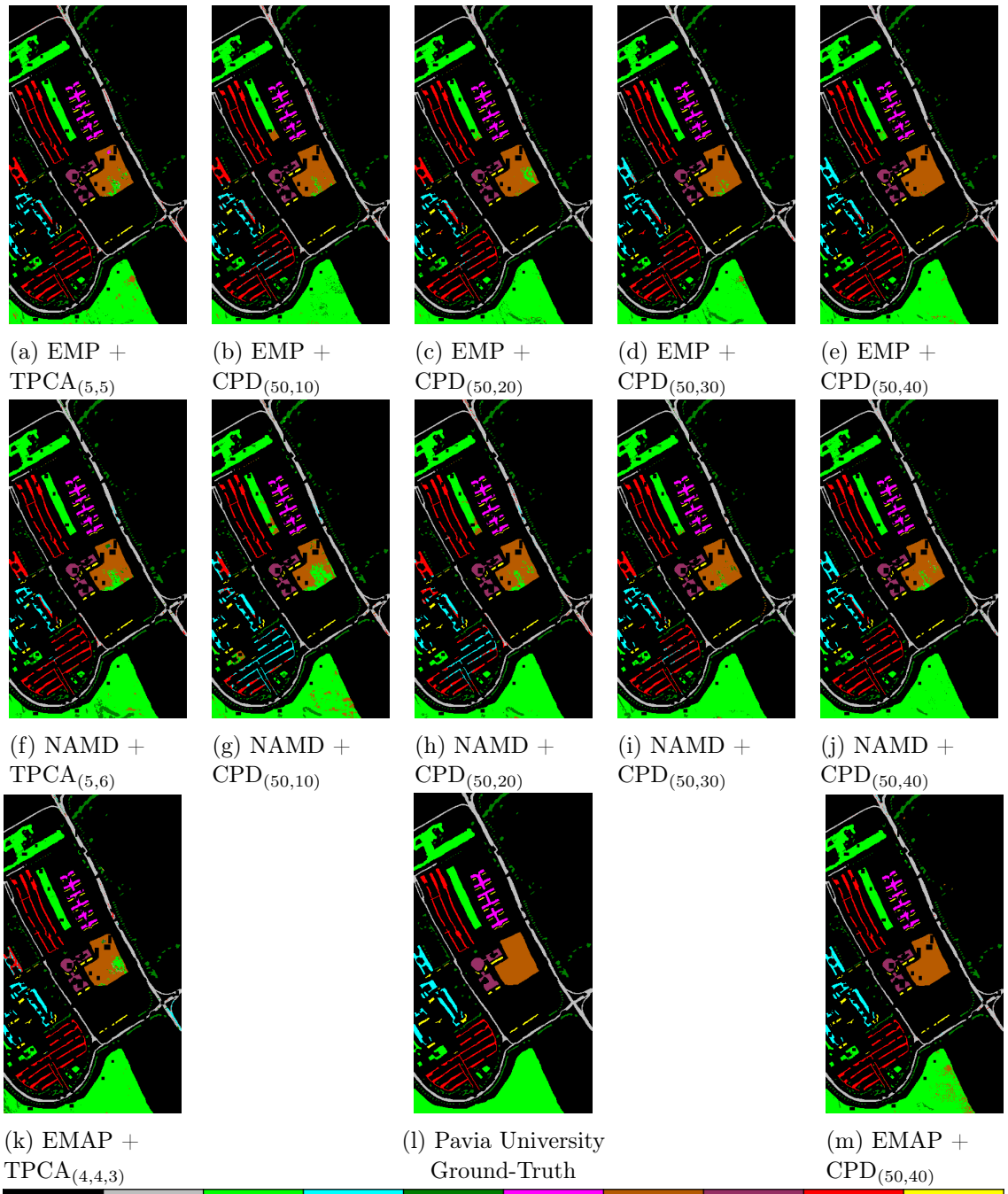
4.4.3.1 EMP

In the case of the DFC image, the set-up and the parameters are almost the same as those of Pavia University. Only the differences are mentioned in the following.

The structuring elements are disk-shaped of different sizes, in pixels: $\{2, 7, 12, 17, 22, 27\}$. This also means that the fourth mode of the tensor is of dimension $K = 13$ with the arrangement shown in (4.2). After merging the first two modes, we obtain the tensor $\boldsymbol{\mathcal{T}} \in \mathrm{R}^{664845\times144\times13}$. After that we obtain the compressed tensor $\boldsymbol{\mathcal{G}} \in \mathrm{R}^{1872\times144\times13}$.

As for the NCPD, we set the values of the rank to $R = \{15, 20, 30, 40\}$. The factor matrices and the parameters of the CPD are initialized similarly to the case in Pavia University. For TPCA, three to five principal components were chosen for the third-mode tensor unfolding explaining the data by 99.60%, 99.72%, and 99.81% respectively, and four to six components were chosen for the fourth-mode tensor unfolding explaining the data by 99.28%, 99.52%, and 99.69% respectively.

Discussion. In Table 4.5, we show some of the results with their reconstruction errors. In this case, we only show the changes in the rank of the CPD, where we notice that as the value of the rank goes higher, the reconstruction error improves, while the classification accuracy doesn't follow the same pattern as before even though it exceeds the values given by TPCA in all cases. The best two overall accuracy values obtained by TPCA are shown in the table in order to keep it consistent with Table 4.6. Additionally, CPD could bring better accuracy results starting at 15 features compared to 30 features for TPCA. Figures 4.11b, 4.11d, 4.11f, 4.11h, and 4.11j show the corresponding classification maps.



(a) EMP + TPCA$_{(5,5)}$ (b) EMP + CPD$_{(50,10)}$ (c) EMP + CPD$_{(50,20)}$ (d) EMP + CPD$_{(50,30)}$ (e) EMP + CPD$_{(50,40)}$

(f) NAMD + TPCA$_{(5,6)}$ (g) NAMD + CPD$_{(50,10)}$ (h) NAMD + CPD$_{(50,20)}$ (i) NAMD + CPD$_{(50,30)}$ (j) NAMD + CPD$_{(50,40)}$

(k) EMAP + TPCA$_{(4,4,3)}$ (l) Pavia University Ground-Truth (m) EMAP + CPD$_{(50,40)}$

(n) Classes Map

Figure 4.10: Classification maps for Pavia University HSI.

Method	No. of Features (R)	Rec. Error %	OA %	AA %
EMP + $\text{TPCA}_{(4,5)}$	20	-	83.20	85.64
EMP + $\text{TPCA}_{(5,6)}$	30	-	83.38	85.93
EMP + $\text{CPD}_{(50,15)}$	15	4.40	<u>84.97</u>	<u>87.04</u>
EMP + $\text{CPD}_{(50,20)}$	20	3.35	<u>84.78</u>	<u>86.87</u>
EMP + $\text{CPD}_{(50,30)}$	30	2.35	<u>84.70</u>	<u>86.80</u>
EMP + $\text{CPD}_{(50,40)}$	40	2.01	<u>84.79</u>	<u>86.97</u>

Table 4.5: DFC. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMP set-up. The features column indicates the size of the feature space in the classification. The best two values given by TPCA are shown. The cases when CPD does better are underlined.

4.4.3.2 NAMD

In the case of NAMD, for TPCA, three to five principal components were chosen for the third-mode tensor unfolding explaining the data by 99.76%, 99.83%, and 99.87% respectively, and four to six components were chosen for the fourth-mode tensor unfolding explaining the data by 95.86%, 98.04%, and 99.21% respectively.

Discussion. In Table 4.6, we show some of the results. In this case, we notice that as the value of the rank goes higher, both the reconstruction error and the classification accuracy improve. Since the best classification accuracy given by TPCA showed a better classification than the ones given by CPD, we included the second best as well. In this case, the classification accuracy results given by CPD are around the range of those given by TPCA. Figures 4.11c, 4.11e, 4.11g, 4.11i, and 4.11k show the corresponding classification maps.

Method	No. of Features (R)	Rec. Error %	OA %	AA %
NAMD + $\text{TPCA}_{(5,6)}$	30	-	83.76	86.10
NAMD + $\text{TPCA}_{(3,6)}$	18	-	<u>84.55</u>	<u>86.83</u>
NAMD + $\text{CPD}_{(50,15)}$	15	15.05	80.79	83.65
NAMD + $\text{CPD}_{(50,20)}$	20	10.77	81.16	83.93
NAMD + $\text{CPD}_{(50,30)}$	30	7.68	82.46	85.16
NAMD + $\text{CPD}_{(50,40)}$	40	5.94	84.07	86.49

Table 4.6: DFC. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the NAMD set-up. The features column indicates the size of the feature space in the classification. The best two values given by TPCA are shown. The cases when CPD does better are underlined.

4.4.3.3 EMAP

Here, we used the same four types of attributes, where the choice of the threshold values was also based on observed changes between transformations:

- Area of the region; $\lambda_a = \{300, 2000, 6000, 9000\}$
- Diagonal of the bounding box; $\lambda_d = \{20, 45, 75, 150\}$
- Standard deviation; $\lambda_s = \{20, 30, 40, 50\}$
- Moment of inertia; $\lambda_i = \{0.2, 0.3, 0.4, 0.5\}$

Consequently, we obtain a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{664845 \times 144 \times 9 \times 4}$.

Following the same steps taken in the EMAP part of Pavia's HSI, and considering the larger block of data in the case of DFC, NMF is performed with a rank of 40, chosen with a reconstruction error of 0.48%. As for the PCA part of the original image, five principal components were chosen, explaining the original HSI with 99.87%.

As for the CPD, we choose a rank of 40. For TPCA, three principal components were chosen for the third-mode tensor unfolding explaining the HSI by 99.15%, six components were chosen for the fourth-mode unfolding explaining the data by 99.68%, and three components were chosen for the fifth-mode unfolding explaining the data by 99.97%.

Table 4.7 shows some of the results. CPD showed a better classification accuracy with 40 features compared to 54 features in the case of TPCA. Figures 4.11l and 4.11m show the corresponding classification maps.

Method	No. of Features (R)	Rec. Error %	OA %	AA %
$\text{PCA}_{(5)}$	5	-	74.12	77.22
$\text{NMF}_{(40)}$	40	0.48	78.83	81.82
EMAP	180	-	83.39	85.78
EMAP + $\text{PCA}_{(15)}$	15	-	82.28	84.96
EMAP + $\text{TPCA}_{(3,6,3)}$	54	-	81.32	84.30
EMAP + CPD$_{(50,40)}$	40	1.95	<u>83.19</u>	<u>85.76</u>

Table 4.7: DFC. Some records of Overall Accuracy (OA) and Average Accuracy (AA), with Reconstruction Error (Rec. Error) in case of CPD, for the EMAP set-up. The features column indicates the size of the feature space in the classification. The best value given by TPCA is shown. The cases when CPD does better are underlined.

Finally, Table 4.8 shows per-class results of Overall Accuracy and Average Accuracy of selected examples from Tables 4.5, 4.6 and 4.7 in a side-to-side comparison including the three morphological set-ups (EMP, AMD, and EMAP) and two decomposition methods (TPCA and CPD). From TPCA to CPD, we notice again that the accuracy of some classes improved or

worsened only slightly. As for significant differences, we notice that some classes are significantly better identified using TPCA than using CPD, but in general it is rather better in the case of CPD. For instance, class Commercial is better identified in the case of NAMD using TPCA than using CPD, while it is better in the cases of EMP and EMAP using CPD than using TPCA.

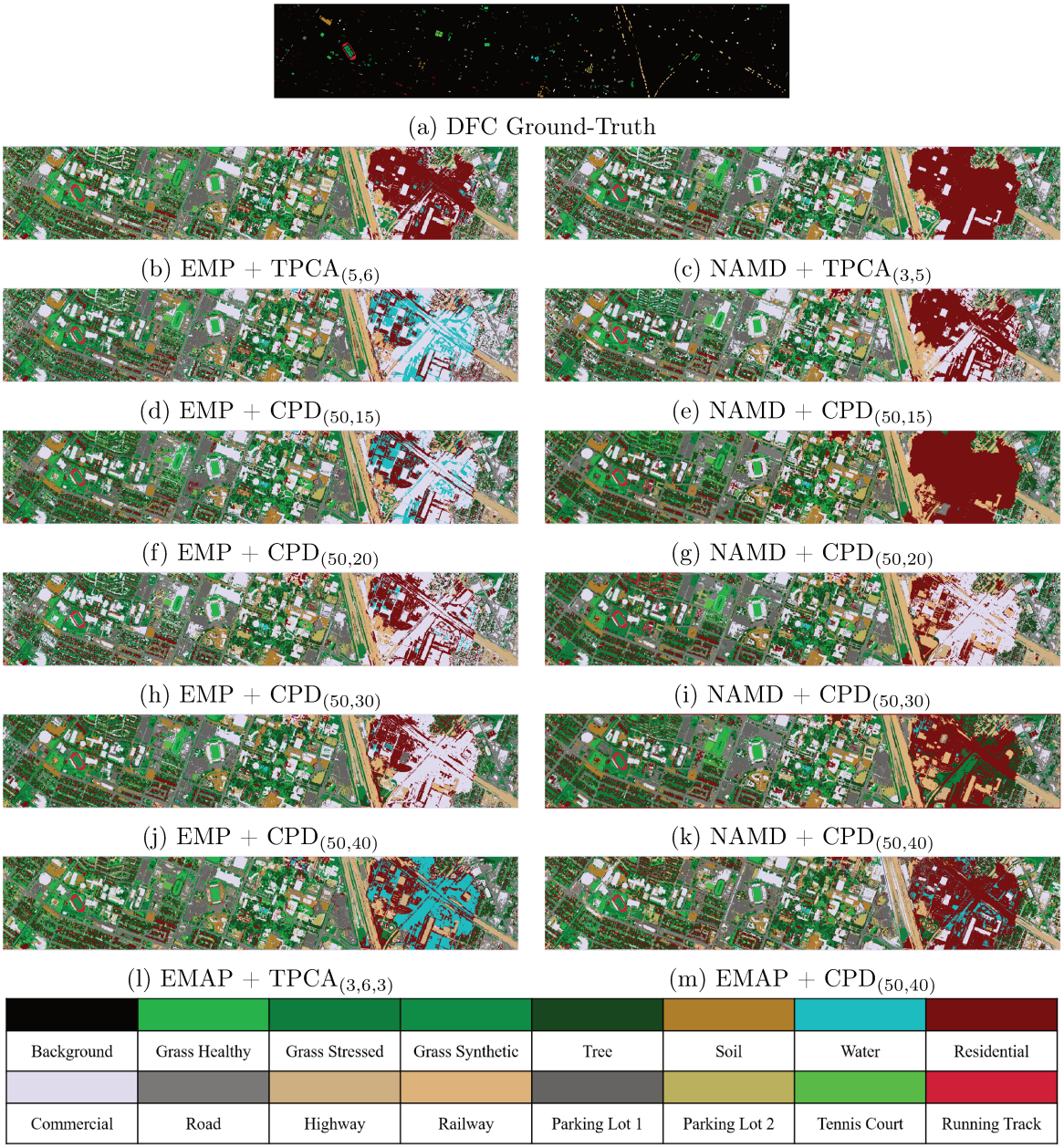
Method	EMP %		NAMD %		EMAP %	
	$\text{TPCA}_{(5,6)}$	$\text{CPD}_{(50,15)}$	$\text{TPCA}_{(3,6)}$	$\text{CPD}_{(50,40)}$	$\text{TPCA}_{(3,6,3)}$	$\text{CPD}_{(50,40)}$
Grass Healthy	81.86	79.58	80.43	79.58	82.24	81.76
Grass Stressed	82.80	84.02	85.05	98.21	82.70	84.30
Grass Synthetic	100	99.60	100	100	100	100
Tree	84.28	90.81	79.45	96.11	87.87	79.92
Soil	98.95	98.67	100	99.52	97.44	99.14
Water	95.10	95.10	95.10	95.10	95.10	95.10
Residential	88.71	79.47	90.67	95.33	78.26	80.69
Commercial	71.03	83.09	70.65	43.96	41.40	75.21
Road	81.01	85.08	82.62	84.79	85.64	75.16
Highway	68.33	68.43	68.24	68.53	68.05	67.56
Railway	82.35	83.87	82.54	84.81	83.20	85.76
Parking Lot 1	73.87	79.73	87.22	69.83	82.99	81.94
Parking Lot 2	80.70	78.59	80.70	82.45	79.64	80.35
Tennis Court	100	100	100	100	100	100
Running Track	100	99.57	99.78	99.15	100	99.57
OA	83.38	84.97	84.55	84.07	81.32	83.19
AA	85.93	87.04	86.83	86.49	84.30	85.76

Table 4.8: DFC. Some per-class accuracy records including those of Overall and Average Accuracies for the EMP and NAMD set-ups.

4.4.4 Unconstrained CPD

The emphasis in this work was on NCPD, which is more intuitive when it comes to physical interpretation of the data, and has an additional potential advantage for semi-supervised classification thanks to its contribution to unsupervised classification (e.g., Chapter 5 and [102]), for which we leave some remarks in Section 4.5. However, we think that classifying the data is still possible without the nonnegativity constraints. The advantage in this case is that the decomposition without constraints is faster than that with constraints.

Considering that CPD without constraints is another way of rewriting the data in a tensor decomposition format, it was worth showing some of the results even though they might have no physical interpretation. In this case, the original tensor is still compressed and the compressed version becomes the input to AO-ADMM. Since there are no constraints on the



(a) DFC Ground-Truth

(b) EMP + $\text{TPCA}_{(5,6)}$ (c) NAMD + $\text{TPCA}_{(3,5)}$

(d) EMP + $\text{CPD}_{(50,15)}$ (e) NAMD + $\text{CPD}_{(50,15)}$

(f) EMP + $\text{CPD}_{(50,20)}$ (g) NAMD + $\text{CPD}_{(50,20)}$

(h) EMP + $\text{CPD}_{(50,30)}$ (i) NAMD + $\text{CPD}_{(50,30)}$

(j) EMP + $\text{CPD}_{(50,40)}$ (k) NAMD + $\text{CPD}_{(50,40)}$

(l) EMAP + $\text{TPCA}_{(3,6,3)}$ (m) EMAP + $\text{CPD}_{(50,40)}$

(n) Classes Map

Figure 4.11: Classification maps for DFC HSI.

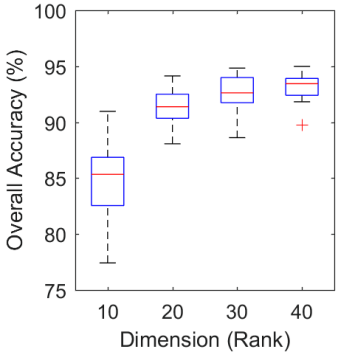
uncompressed factors of the decomposition, AO-ADMM becomes pretty much the same as computing Alternating Least Squares. We computed 20 realizations of unconstrained CPD for each value of the rank, $R = \{10, 20, 30, 40\}$ in the case of Pavia University and $R = \{15, 20, 30, 40\}$ in the case of DFC. We note that, practically, computing the CPD yields some variations between the factor matrices that are obtained from different initializations, which are drawn randomly as described in the previous section (except that in the case of unconstrained CPD, we don't force the initializations to be in the positive quadrant).

First, we show, for each of the setups, the overall accuracy and reconstruction error results of 20 realizations in the box plots of figures 4.12 and 4.14 representing the HSI of Pavia University and DFC respectively, followed by the tables 4.13 and 4.15 respectively that show specific values of overall accuracy from the box plots; the lower adjacent, median and upper adjacent of the boxes, and the overall accuracy corresponding to the minimal reconstruction error among the 20 realizations. We notice in most of the cases a general increase in the overall accuracy and a likewise decrease in the reconstruction error as the rank increases, and in most of the cases the results are comparable to, or higher than, those found in tables 4.1, 4.2, 4.5 and 4.6 especially with higher values of the rank. The only exception to the latter is in the AMD setup of the DFC image where the values of the overall accuracy fluctuate a bit, which could be caused by a bad modeling of the classifier since cross-validation doesn't always find the most optimized solution for the hyper-parameters of SVM, but by looking at Table 4.15, we notice that the range of the values is still comparable to a state-of-the-art method such as TPCA.

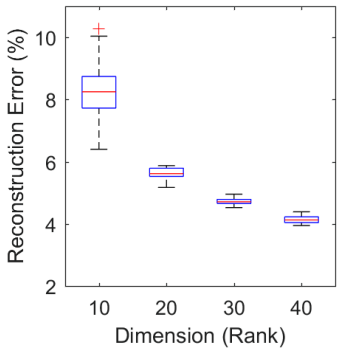
4.5 Concluding Remarks

In this paper, CPD was proposed as a dimensionality reduction technique in the framework of *supervised* spectral-spatial classification of hyperspectral data with spatial information being added using Mathematical Morphology as an application. In general, CPD provides an intuitive approach to deal with such multi-modal feature data where the results in the decomposition can be used for pixel-wise classification in a low dimensional feature space without loss of information. The focus here was on NCPD (with big data compression), which further provides a better understanding of the results that are distributed over the decomposed factors, and where each factor describes one of the modes of the tensor in an interpretable way. Experiments on two datasets and three different morphological settings were carried out to explore the classification aspect of the proposed technique, compared to results obtained through TPCA as a base algorithm in the framework of tensor modeling. The effect of some parameters such as the number of iterations, the rank of the decomposition, the reconstruction error, and the use of constraints was explored. The classification results were promising to say that the data found in the first factor matrix of the CPD can be seen as pixel data and hence classified.

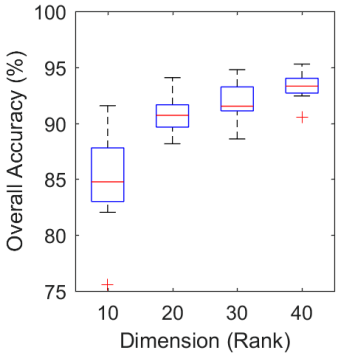
This kind of application allows to go further. For instance, one way would be to explore the potential unmixing aspect of using NCPD of hyperspectral data based on multi-modal feature study (spectral and spatial features for instance) in the framework of *unsupervised* classification or blind separation. This would better highlight (a) both the use of CPD compared to other decomposition methods that have been used, and (b) that of nonnegativity constraints. This application is explored in details in Chapter 5 including a study on "Extended Linear Mixing Model" for hyperspectral unmixing problems in the case of tensors. Finally, we note that there are some challenges for using the compressed NCPD. First, it may suffer from long-time execution due to per-iteration compression and decompression steps. Second, due to the iterative nature of its approximation, it may fall into local minima.



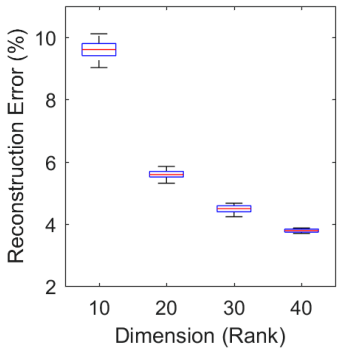
(a) EMP: overall accuracy with respect to different values of the rank



(b) EMP: reconstruction error of the CPD with respect to different values of the rank



(c) AMD: overall accuracy with respect to different values of the rank

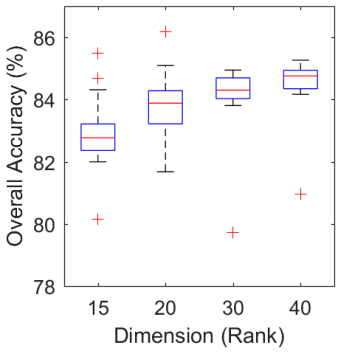


(d) AMD: reconstruction error of the CPD with respect to different values of the rank

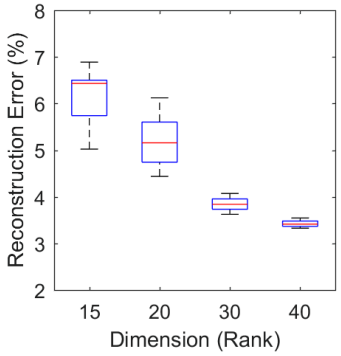
Figure 4.12: Pavia University. Box plots of overall accuracy (left) and reconstruction error (right) with respect to the rank of the CPD. Figures 4.12a-4.12b correspond to EMP and figures 4.12c-4.12d correspond to AMD. Each box represents 20 decompositions, each carried with a different initialization along with the corresponding value of the rank. Remarkable values of overall accuracy are shown in Table 4.13.

EMP	$R = 10$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	77.42	88.06	88.61	91.82
Median	85.34	91.38	92.62	93.43
Upper Adjacent	90.96	94.13	94.83	94.98
Minimal Rec. Error	85.25	94.13	92.60	94.42
AMD	$R = 10$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	82.03	88.16	88.59	92.43
Median	84.76	90.70	91.51	93.31
Upper Adjacent	91.56	94.07	94.78	95.28
Minimal Rec. Error	91.56	89.47	93.96	95.28

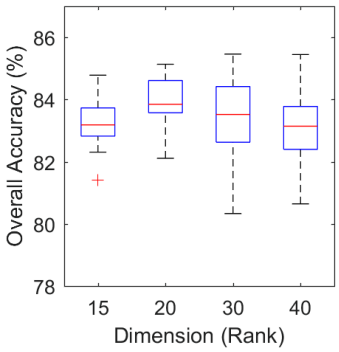
Figure 4.13: Pavia University. Values of overall accuracy taken from the boxes of Figure 4.12.



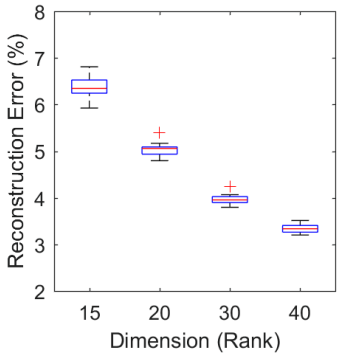
(a) EMP: overall accuracy with respect to different values of the rank



(b) EMP: reconstruction error of the CPD with respect to different values of the rank



(c) AMD: overall accuracy with respect to different values of the rank



(d) AMD: reconstruction error of the CPD with respect to different values of the rank

Figure 4.14: DFC. Box plots of overall accuracy (left) and reconstruction error (right) with respect to the rank of the CPD. Figures 4.14a-4.14b correspond to EMP and figures 4.14c-4.14d correspond to AMD. Each box represents 20 decompositions, each carried with a different initialization along with the corresponding value of the rank. Remarkable values of overall accuracy are shown in Table 4.15.

EMP	$R = 15$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	81.99	81.67	83.80	84.16
Median	82.77	83.88	84.30	84.75
Upper Adjacent	84.31	85.09	84.94	85.26
Minimal Rec. Error	85.49	83.95	84.16	84.16
AMD	$R = 15$	$R = 20$	$R = 30$	$R = 40$
Lower Adjacent	82.30	82.11	80.33	80.46
Median	83.18	83.84	83.52	83.14
Upper Adjacent	84.78	85.13	85.46	85.44
Minimal Rec. Error	83.47	83.57	82.69	82.97

Figure 4.15: DFC. Values of overall accuracy taken from the boxes of Figure 4.14.

Chapter 5

Hyperspectral Multi-feature Unmixing Through Tensor Decomposition

Sommaire

Abstract

Following Section 2.5.2 and Chapter 4, in this chapter, hyperspectral unmixing is explored using Canonical Polyadic (CP) decomposition based on a multi-feature study taking into account not only the spectral diversity, but also an additional physically distinctive feature diversity extracted from the image. By representing them as tensors spanning modes of pixels, spectral features, and third-diversity features, matrix models become insufficient. Moreover, CP decomposition is also defined in the context of Extended Linear Mixing Model (ELMM), which accounts for spectral variabilities present in the data. This work involves a deeper mathematical, physical, and graphical interpretation of ELMM for tensors and demonstrates its advantages by incorporating Mathematical Morphology similarly to Chapter 4. The findings are quantitatively and qualitatively evaluated through synthetic and real hyperspectral data and assessed from two perspectives: the quality of extracted materials compared to related state-of-the-art, and the interpretation of how spectral variability is modeled compared to patch-tensor unmixing.

5.1 Introduction

In Chapter 4, we talked about an application of supervised classification using CP decomposition (CPD) and mentioned that CPD can be used in an application of blind source separation (BSS). In this chapter, we talk about Hyperspectral Unmixing (HU) [10], which is a case of BSS for hyperspectral images. In Section 2.5, we gave some background on HU and talked about some of the works that involve matrix and tensor factorization. Among these works, we mentioned a contribution of this thesis that explores and generalizes the application of multi-feature unmixing and its importance in terms of tensor decomposition and ELMM with mathematical and physical interpretations. In this chapter, we dive into the details and show the experiments and results. The rest of this section recalls and continues from Section 2.5.2 and serves as an introduction to the following sections.

An important property of HSIs is that they contain pure and mixed pixels (see Figure 5.1). Depending on the spatial resolution of the hyperspectral sensor, a pure pixel is one that covers an area of the scene made of one type of material and reflects its spectral signature, while a mixed pixel is one that covers an area containing multiple types of materials (whose appearance may be homogeneous or heterogeneous) and reflects a mixed spectral signature. HU [10] refers to the process of unmixing (separating) the HSI data into its fundamental spectral composition, i.e., with respect to the spectral signatures of pure materials, also called *endmembers*, which allows to understand and classify the different components of a scene.

Here, we are interested in HU as a tensor-based model in general terms (i.e., not restricted to MM) where HSIs are represented with additional diversities. To begin with, and without loss of generality, it is evident that having additional information gives more context on any type of data and their distinctive features and behaviors, thus improving the *blind* identification of the latent fundamental materials. As mentioned in Section 2.5.2.1, the majority of the HU literature assumes that a HSI is represented as a matrix in order to fit for NMF techniques, which could become insufficient for future applications of high-order data processing especially in the presence of additional diversities. One way to overcome these limitations is through tensor analysis, with advantages for HU:

- Unique solutions under mild conditions, thus requiring less dependence on *a priori*

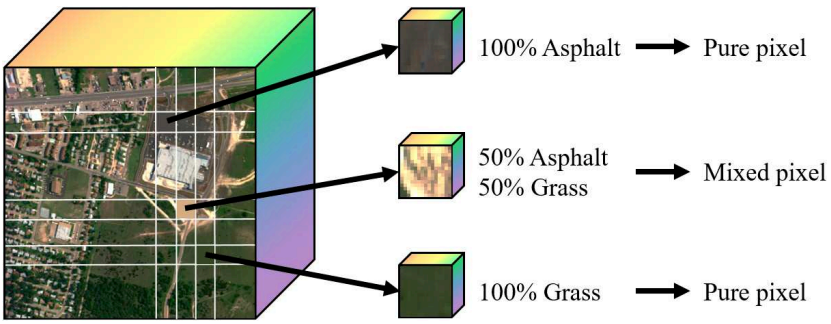


Figure 5.1: Mixed and pure pixels of a HSI

knowledge.

- Joint and intuitive multi-modal processing, which allows to naturally incorporate multiple modes of additional information *within* the process of unmixing in order to simultaneously extract materials with said multi-feature properties.

Another advantage that is explored as part of this work is that it is possible to achieve a unique unmixing of HSIs using features extracted from the data itself, as well as improving the identification of the materials, which is the case of Mathematical Morphology (MM) [57] as shown in Chapter 4.

Motivated by the importance of high-order data and the potential of tensor-based HU in jointly exploring multiple features (including the spectral ones), in this work, we propose a *general methodological study* of hyperspectral multi-linear unmixing, which, at its base, joins and extends the following points in the literature: (a) Blind feature extraction from high-order data (inspired from [102]), (b) Dealing with *spectral variability* in HU through tensor modeling (inspired from [103]), and (c) Exploiting the multi-modal relationships as much as possible, with an emphasis on the role of additional diversities that are built using physically meaningful transformations of the original HSI, with an example of MM, which has been previously successfully implemented in HSI scene classification [56, 58, 57]. That said, before expanding on the context of this study, we summarize its contributions as follows:

- Inspired by [103], we focus and expand on the relationship (5.4) between tensor CPD and the ELMM model in more general terms by showing that it provides a common theoretical ground for the aforementioned topics and generalizes the way HU is interpreted for high-order multi-feature data, i.e., regardless of the nature of the third-mode diversity. For that, we also provide graphical and visual representations of the problem.
- We take advantage of this generalization to show the importance of unmixing high-order HSIs that jointly incorporate multiple features of physically meaningful properties, one of which is extracted from the original HSI. For example, in the case of MM, materials are blindly identified based on spectral *and* spatial/morphological properties (e.g. size and brightness of objects) instead of only the spectral ones.
- We explain in full details the connection between CPD and the problem of spectral variability for HU. This is complementary to [103], where the connection was briefly described and focused on the case of building a third-order HSI tensor from neighborhood patches. In this regard, the significance is two-fold, abstract and applicative:
  - Abstractly speaking, this allows to easily draw the connection between the functions (or parameters) that construct the third-mode diversity (not restricted to patches) and the expected quality and quantity of the extracted materials (including how the spectral variability functions are influenced and defined).
  - In terms of application, the choice of MM can be better to deal with spectral variability than that of patches because MM promotes that pixels are selectively connected to their neighbors based on shared spatial properties, which gives the

data awareness of pixels connected as spatial objects of the same material, while patches promote indifferent stacking of pixels within a spatial patch where the pixels can correspond to different materials. Briefly speaking, in the case of MM, CPD smooths out the spectral variabilities within similar materials.

- We propose a way to incorporate Abundance Sum-to-one Constraints (ASC) in the computation of third-order CPD using the algorithm AO-ADMM.

At this stage, it is important to note that multi-linear unmixing can be slightly different from classical spectral unmixing due to the involvement of additional information and the nature of the results where, in general, multiple patterns are being observed and could influence the result. For instance, compared to classical spectral unmixing, an additional diversity may lead to a more specific separation of the materials of the scene thanks to incorporating more information on the data. In the following, we expand on the related works and our proposal.

5.1.1 Motivation and Related Works

A direct approach to HU is the classical linear modeling of the problem, which is known as the Linear Mixing Model (LMM) and can be directly posed as a NMF problem [10]. Figure 5.2 shows an illustration of the LMM problem through NMF: For a HSI matrix $\boldsymbol{M} \in \mathbb{R}^{I \times J}$ containing I pixels (as rows) and J spectral bands (as columns), LMM assumes that each pixel row $\boldsymbol{m}_i \; \forall \, i \in \{1, \ldots, I\}$ can be expressed as the linear combination of R endmembers $\boldsymbol{b}_r$:

$$\boldsymbol{m}_i = \sum_{r=1}^{R} a_{ir} \boldsymbol{b}_r \tag{5.1}$$

such that all the values are *nonnegative* and that $\sum_{r=1}^{R} a_{ir} = 1 \; \forall \, i \in \{1, \ldots, I\}$, where the values of a_{ir} stand for the *fractional abundances* of the endmembers (the proportions of endmembers in each pixel). The latter constraint is called the *Abundance Sum-to-one Constraint* (ASC). This model is reformulated as $\boldsymbol{M} = \boldsymbol{A}\boldsymbol{B}^{\mathsf{T}}$, subjected to nonnegativity and ASC. Consequently, HU can be carried out by finding the two nonnegative matrices $\boldsymbol{A} \in \mathbb{R}^{I \times R}$, whose rows represent the fractional abundances at each pixel, and $\boldsymbol{B} \in \mathbb{R}^{J \times R}$, whose columns represent the estimated endmembers [10]. Nonnegativity of the factors ensures a part-based additive representation since fractional abundances and spectral signatures are nonnegative, and ASC ensures that the endmember composition sums to one for each pixel so that the pixels lie inside a convex hull formed by the endmembers. Sometimes, sparsity is imposed assuming that a pixel area represents a combination of only few endmembers, which also reduces the problem of solution uniqueness in NMF [25, 111].

While LMM is seen as a direct linear model for HU, it is not enough to model real-life HSIs due to the presence of spectral variabilities represented by nonlinear effects or illumination conditions. One way to account to said effects is through Extended LMM (ELMM) [39], which in general assumes additional degrees of freedom that account to the existing spectral variabilities at the pixel level by introducing a pixel-dependent variability function $\boldsymbol{f}_i : \mathbb{R}^J \rightarrow$

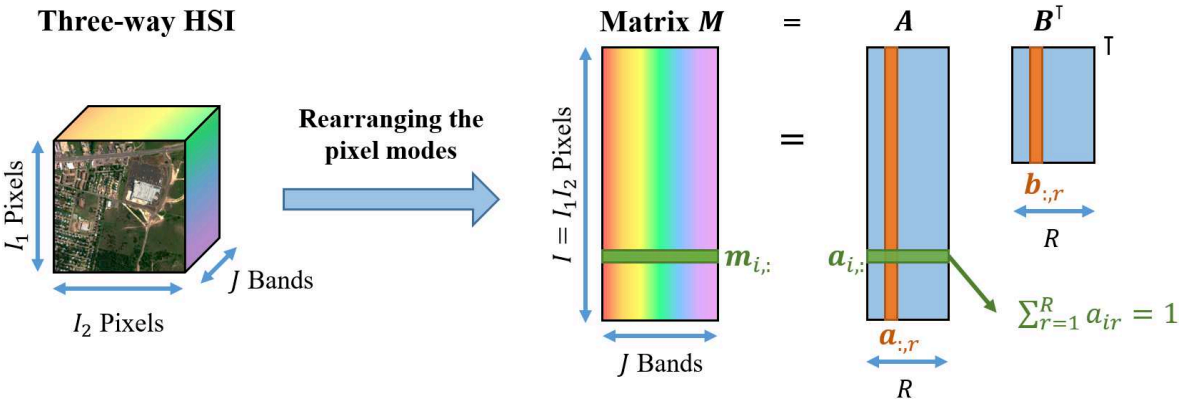


Figure 5.2: Classical matrix-based HU using NMF (i.e., Linear Mixing Model)

$\mathbb{R}^J$, which maps each endmember $\boldsymbol{b}_r$ to a new spectral signature $\boldsymbol{b}_r^{(i)}$ that best reflects the targeted variabilities:

$$\boldsymbol{m}_i = \sum_{r=1}^{R} a_{ir} \boldsymbol{f}_i(\boldsymbol{b}_r) = \sum_{r=1}^{R} a_{ir} \boldsymbol{b}_r^{(i)}. \tag{5.2}$$

Dealing with spectral variability in HU using ELMM is an active topic that has seen a lot of progress recently [104, 42, 40, 41, 103, 59]. Currently, the approaches can be put into three categories depending on how $\boldsymbol{f}_i$ is modeled [39]: (a) Spectral Bundles [104], (b) Computational Models, and (c) Parametric Models including physical ones (e.g. Hapke model) [40] and Constrained Least Squares (CLS) [104]. These models are usually incorporated in non-linearly *constrained* NMF situations, but lately, a relationship was made between tensor CPD and ELMM [103], i.e., expression (5.4), which becomes the core of our proposal by focusing on its theoretical aspect and potential in generalizing multi-linear unmixing.

CPD and ELMM. Figure 5.3 shows an illustration of CPD of a third-order tensor with spectral and spatial features. For a third-order tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ of rank R, CPD aims at finding the three factor matrices $\boldsymbol{A} \in \mathbb{R}^{I \times R}$, $\boldsymbol{B} \in \mathbb{R}^{J \times R}$, and $\boldsymbol{C} \in \mathbb{R}^{K \times R}$, such that:

$$\boldsymbol{\mathcal{T}} = \boldsymbol{\Lambda} \underset{1}{\bullet} \boldsymbol{A} \underset{2}{\bullet} \boldsymbol{B} \underset{3}{\bullet} \boldsymbol{C} \;\leftrightarrow\; \boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r \;\leftrightarrow\; t_{i,j,k} = \sum_{r=1}^{R} a_{i,r}\, b_{j,r}\, c_{k,r} \tag{5.3}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{R \times R \times R}$ is a diagonal tensor of ones, and R is the number of extracted materials. Here, CPD is a very suitable tool to separate third-order hyperspectral data having spectral and spatial feature modes into three factor matrices: $\boldsymbol{A}$ for fractional abundances, $\boldsymbol{B}$ for spectral features, and $\boldsymbol{C}$ for spatial ones (Figure 5.3). Now, as shown in [103], assuming that $\boldsymbol{T}_{:,:,k}\ \forall\ k \in \{1, \ldots, K\}$ represents the k-th frontal slice of $\boldsymbol{\mathcal{T}}$, one can also write:

$$\boldsymbol{T}_{:,:,k} = \boldsymbol{A}\,\mathrm{Diag}\{\boldsymbol{c}_{k,:}\}\boldsymbol{B}^{\mathsf{T}} = \boldsymbol{A}\boldsymbol{\Psi}_{(k)}\boldsymbol{B}^{\mathsf{T}} = \boldsymbol{A}\tilde{\boldsymbol{f}}_k(\boldsymbol{B})^{\mathsf{T}} \tag{5.4}$$

$$\Longleftrightarrow\ \boldsymbol{t}_{i,:,k} = \sum_{r=1}^{R} a_{ir}\ (\boldsymbol{b}_r c_{kr}) = \sum_{r=1}^{R} a_{ir} \boldsymbol{f}_k(\boldsymbol{b}_r) \tag{5.5}$$

These two expressions resemble (5.2), but the major difference is that ELMM becomes dependent on full frontal slices instead of independent pixels, and this is where the interpretation of

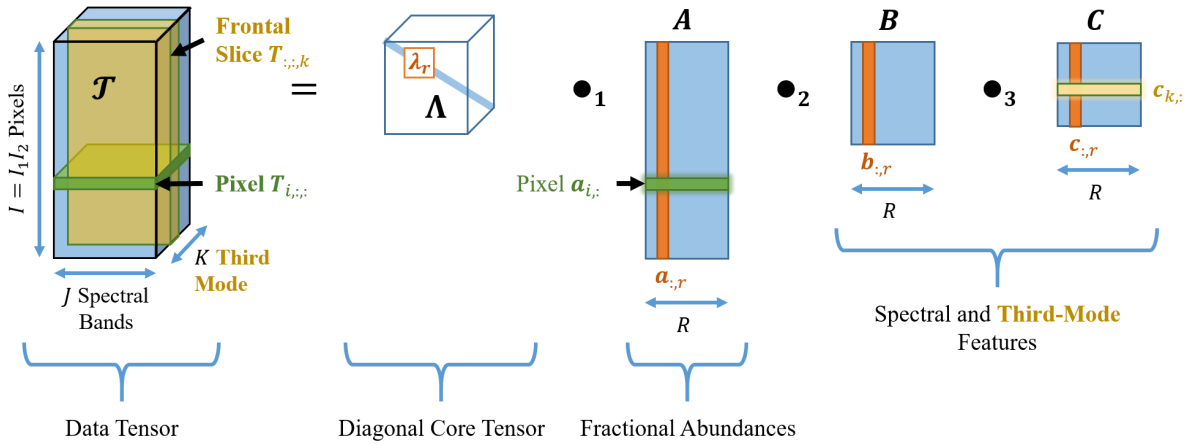


Figure 5.3: An illustration of third-order tensor CPD where $\mathcal{T}$ is the data tensor. One pixel of $\mathcal{T}$ represents a matrix of features (horizontal green slab) and associates to a row in $\boldsymbol{A}$ at the same i-th index. Similarly, one slice of $\mathcal{T}$ (frontal yellow slab) represents a transformation of the HSI and associates to a row in $\boldsymbol{C}$ at the same k-th index.

multi-feature HU is generalized. Looking at (5.4), the frontal slices, which may represent transformations of the original HSI [57] or temporal evolution of a scene [102], i.e. $\boldsymbol{T}_{:,:,k} = \boldsymbol{h}_k(\boldsymbol{M})$, have a direct influence on the spectral variability function and the way spectral variability is modeled, i.e. $\boldsymbol{f}_k$ or $\boldsymbol{\Psi}_{(k)}$, which is simply reflected as scaling factors in each row of $\boldsymbol{C}$, i.e. $\boldsymbol{c}_{k,:}$. As a result, each frontal slice is inherently factorized into a set of R scaled endmembers where the endmembers themselves (the columns of $\boldsymbol{B}$) are free of the frontal slices. For instance, the endmembers of $\boldsymbol{T}_{:,:k}$ are the set of vectors $\{c_{kr}\boldsymbol{b}_r\}\ \forall\ r \in \{1, \ldots, R\}$, which also means that R controls the number of extracted materials including the scaling factors, and the latter represents the *evolution of materials* across the third mode for each frontal slice on the endmembers. Intuitively, one can say that the effects of the transformations applied on an image, or the natural evolution of a scene, on the fundamental materials can be linearly visible through a tensor decomposition.

This version of ELMM, referred to as Regularized ELMM (RegELMM), was used in [103] in order to impose a spatial regularization on the fractional abundances within a patch of neighborhood pixels, rearranged as a tensor representation as shown in Figure 5.4, which, through a low-rank CPD, smooths the variabilities of the spectral signatures of the pixels that belong to the same patch. A main advantage of considering patch-HSI tensors is that, compared to NMF where the spatial information is completely ignored, it would be possible through a tensor representation to incorporate neighborhood information that smooths out the spectral variabilities and maintain a low-rank decomposition.

Here, we note that while the concept of patches is efficient, it still ignores the physical spatial properties of connected pixels, that is, the spectral variability is equally regulated between pixels of a patch that may contain different types of materials, i.e., possessing different spectral properties. For that, as an application, one way to consider such physical properties is through Mathematical Morphology (MM), which has been successfully implemented in HSI

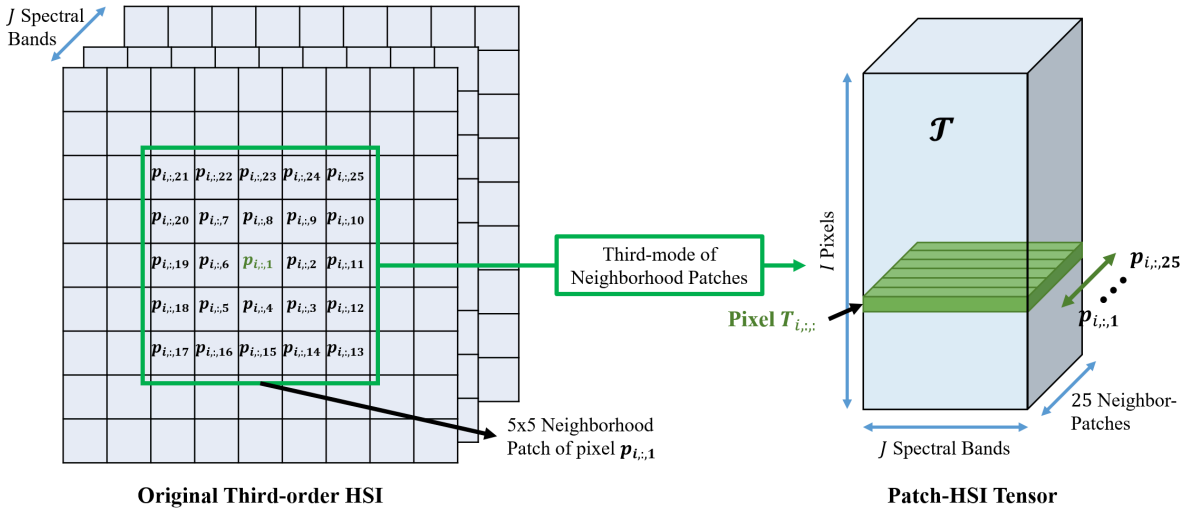


Figure 5.4: Building a HSI tensor from neighborhood patches [103].

scene classification [56, 58, 57], where additional morphological features were incorporated as a new diversity based on different size and brightness levels of the objects of the scene which improved the classification accuracy of the materials. Compared to neighborhood patches, morphological features promote dealing with spectral variability within groups of pixels sharing similar spatial properties. Here, we emphasize on the role of additional diversities that are built using spatial transformations of the original HSI that add physical significance on the objects of the image. As such, building a tensor from neighborhood patches does not contribute new physical input to the data, while building one from morphological transformations induces knowledge of the size of objects and their brightness level in the process of unmixing, which means that the number of extracted features can increase in the case of MM as the process becomes more specific.

Low-rankness assumption of patch-HSI tensors. The claim that patch-HSI tensors do not add new information to the data set appears when we look at the frontal slices individually. If we suppose by definition that each pixel is followed by its neighboring ones along the third mode as shown in Figure 5.4, then the frontal slices are only spatially-shifted versions of $\boldsymbol{M}$, where $\boldsymbol{T}_{(1)} = \boldsymbol{M}$ (we refer to Figures 5.15 and 5.23 for examples of frontal slices of real patch-HSI tensors). Since the patches are relatively very small in size compared to the dimensions of real images, then the difference between the frontal slices is negligible. Recalling the direct relationship between $\boldsymbol{T}_{(k)}$ and $\boldsymbol{f}_{(k)}$ (5.4), we can deduce that the set of spectral variability functions $\forall\, k \in \{1, \ldots, K\}$ reflects the same information. As such, one can assume that the approximation rank of a tensor built from patches is almost the same as that of $\boldsymbol{M}$ (i.e. the same amount of underlying information).

In this sense, one expects that the approximation rank of a tensor built using MM can be higher than that of $\boldsymbol{M}$ since we have more context on the data related to their sizes and brightness levels. However, one can also choose the rank to be as much as that of $\boldsymbol{M}$. We observe that the former case promotes having a more distinctive separation of the materials with spectral and morphological patterns, while the latter promotes spectral smoothing of

variabilities based on a spatial-morphological regularization of the abundances.

5.1.2 Proposal Overview

In this work, we set a general framework for multi-feature HU as high-order data are becoming more accessible nowadays. Under this framework, we have a HSI tensor with a third diversity of physically distinctive features. Knowing that the second diversity represents the spectral one, decomposing such tensors using CPD provides abundance maps (i.e. columns of $\boldsymbol{A}$) with double-feature tags based on the patterns found in the columns of $\boldsymbol{B}$ and $\boldsymbol{C}$. As such, in terms of spectral variability, $\boldsymbol{C}$ can be viewed as a matrix of scaling factors in its rows describing the relevance of the endmembers to the frontal slices, and in terms of unmixing, it can be viewed as third-mode physical patterns in its columns describing each of the materials.

In practice, we choose to build the tensor through MM, where we utilize the so-called Extended Morphological Profile (EMP) [6, 4] in order to build the tensor, after which we perform the joint spectral-morphological unmixing step using Nonnegative CPD (NCPD) and enjoy its properties of uniqueness and low-rankness. In Section 5.2, we further discuss the relationship between CPD and ELMM in a new light through graphical and visual representations, and we compare between patch-HSI and MM-HSI tensors as examples in this regard. Finally, we carry out a series of experiments on synthetic and real HSI data using NMF, patch-HSI NCPD, and MM-HSI NCPD. Through these experiments, we observe the differences between the techniques themselves, give some remarks regarding the influence of constraints, and highlight the manifestation of ELMM in the results. In order to carry out NCPD, we use the algorithm "Alternating Optimization - Alternating Direction Method of Multipliers" (AO-ADMM) [53], and propose a way to implement ASC in AO-ADMM for third-order tensors.

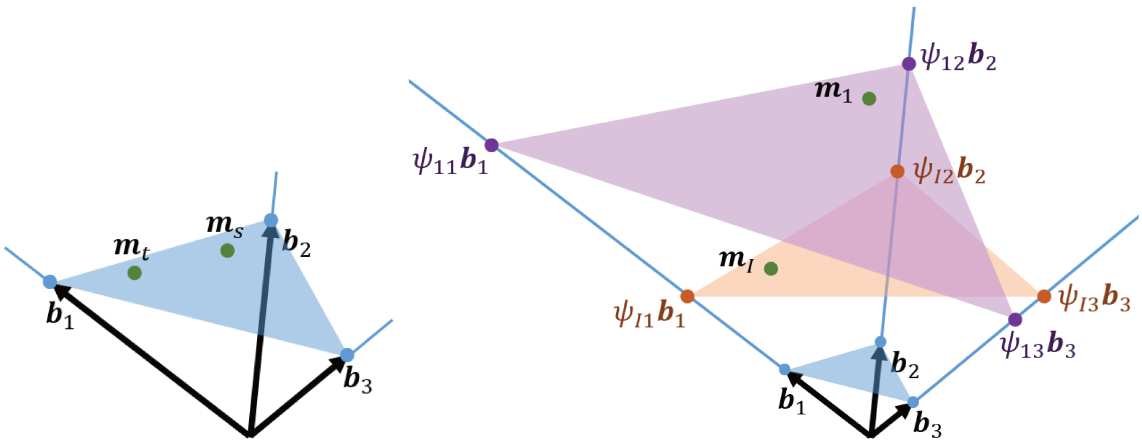
The rest of the chapter is organized as follows. In section 5.2, we talk about CPD and ELMM. In section 5.3, we immediately talk about ASC and AO-ADMM for CPD knowing that CPD has been fully introduced in earlier chapters. In section 5.4, we present the experiments carried out on a synthetic HSI and two real HSI data sets. Finally, we draw some conclusions in section 5.5.

5.2 Extended Linear Mixing Model

In this section, we elaborate on the relationship between NCPD and ELMM. First, we start with a brief account on matrix-based ELMM with graphical and visual representations. After that, we elaborate on the tensor-based RegELMM and compare its functionality to that of the matrix-based model. Finally, we consider two examples of RegELMM where the third modalities are patches and MM, which showcases some of how ELMM manifests with HSI tensors and how MM differs from patches in terms of physical significance. We note that no additional modalities are considered in ELMM. Moreover, RegELMM is an equivalent model to NCPD; it is not restricted to a specific additional modality, but we the term is used to refer

to its tensor-based ELMM-like functionality for HU.

5.2.1 Classical ELMM



(a) LMM (b) ELMM (CLS)

Figure 5.5: Graphical representations of LMM and ELMM (CLS) in the case of three spectral signatures and two sample pixels. The relative coordinates of the two pixels in the convex hulls (i.e. their fractional abundances) are assumed the same across the two models.

Starting with the definition of LMM (5.1), having nonnegativity and ASC suggests that all the pixels lie inside a convex hull whose edges are defined by the set of endmembers. Moreover, the fractional abundances define the graphical position of the pixels in the convex hull. By solving (2.6) under the same constraints, we impose that all the pixels lie on a convex hull defined by the spectral signatures found in the columns of $\boldsymbol{B}$ as illustrated in the case of three spectral signatures $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$ and two sample pixels $\{\boldsymbol{m}_s, \boldsymbol{m}_t\}$ in figure 5.5a.

To deal with the problem of spectral variabilities in HSI that the linearity of LMM ignores, ELMM was proposed as briefly and generally defined in (5.2), Here, the most interesting model with respect to RegELMM in terms of mathematical structure is that of CLS, which is an abstract model where the endmember variability is modeled by a pixel-dependent scaling factor ψ_{ir} as follows:

$$\boldsymbol{m}_i = \sum_{r=1}^{R} a_{ir} \boldsymbol{f}_i(\boldsymbol{b}_r) = \sum_{r=1}^{R} a_{ir} \boldsymbol{b}_r^{(i)} = \sum_{r=1}^{R} a_{ir} \psi_{ir} \boldsymbol{b}_r. \tag{5.6}$$

We note that the term $a_{ir}\psi_{ir}$ is left unfused in order to reflect the similarity between ELMM and NCPD in retrospect. Accordingly, on a graphical representation, the new spectral signatures $\boldsymbol{b}_r^{(i)}$ are allowed to move only along the directions of those of $\boldsymbol{b}_r$, and since it is pixel-dependent, each pixel has its own convex hull representation (i.e. there are as many convex hulls as pixels) as illustrated in the case of three spectral signatures $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$ and

two sample pixels $\{\boldsymbol{m}_s, \boldsymbol{m}_t\}$ in figure 5.5b. Moreover, CLS can also be written as:

$$\boldsymbol{M} = (\boldsymbol{A} \boxdot \boldsymbol{\Psi})\boldsymbol{B}^{\mathsf{T}} \tag{5.7}$$

$$\boldsymbol{m}_i = \boldsymbol{a}_i \boldsymbol{\Psi}_{(i)} \boldsymbol{B}^{\mathsf{T}} \tag{5.8}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{I \times R}$ is the matrix that stores the scaling factors, $\boldsymbol{\Psi}_{(i)}$ is the diagonal matrix formed from the i-th row of $\boldsymbol{\Psi}$, and $\boxdot$ is the Hadamard product. Accordingly, the two representations can be visualized as shown in figure 5.6. These mathematical, graphical, and visual representations will be the key to showcase the RegELMM model, noting that here we have only one data matrix $\boldsymbol{M}$ (compared to several frontal slices in the tensor case) and the focus is at the pixel level (compared to the level of whole frontal slices in the tensor case).

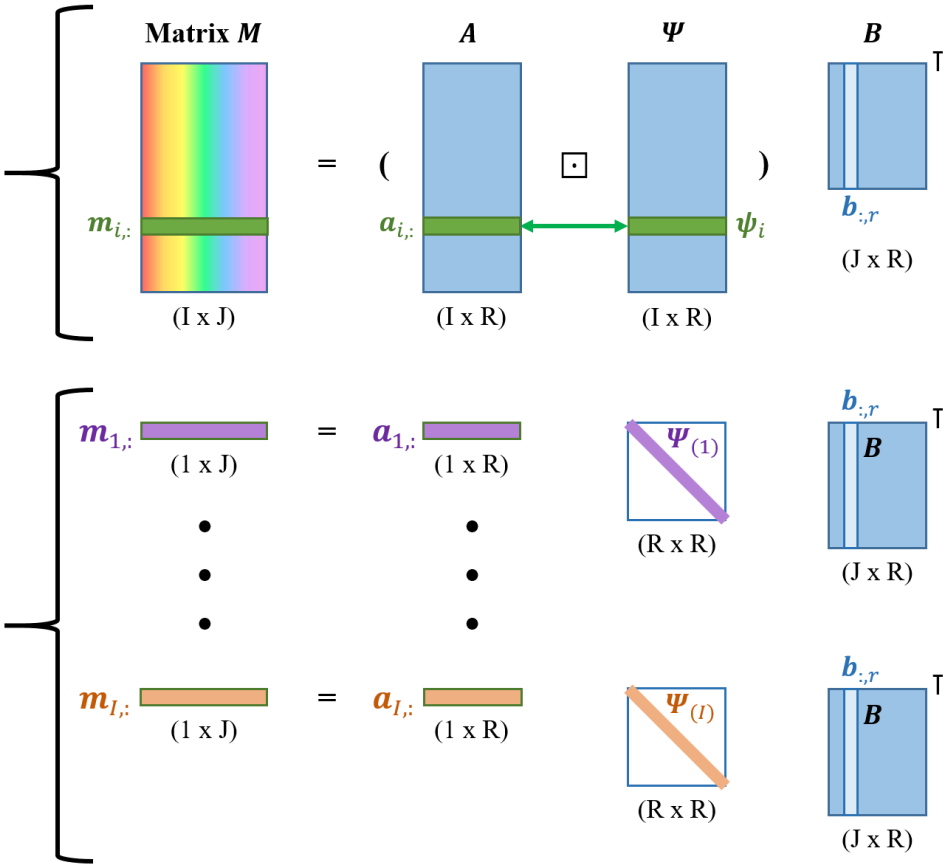


Figure 5.6: Visualization of equations (5.7) (top, using Hadamard product) and (5.8) (bottom, using matrix product). $\boldsymbol{\Psi}_{(i)}$ is the diagonal matrix formed from the i-th row of $\boldsymbol{\Psi}$. The color code of the bottom part follows that of figure 5.5b.

5.2.2 Contribution: Tensor-based ELMM

Here, we talk about RegELMM in general terms (regardless of the application) comparing it to classical ELMM, then we talk about the cases of patches and MM as two of its examples.

In [103], CPD was defined as an equivalent model to a regularized version of ELMM (RegELMM) with an application on patch-HSI tensors. First, the definition stems from the CPD representation shown in equation (5.4) for one frontal slice and equation (5.5) for one pixel of the frontal slice. This indicates that each row of $\boldsymbol{C}$ is supposed to represent the scaling factors at the level of its corresponding frontal slice. Accordingly, the expressions are visualized as shown in figure 5.7. Moreover [103], RegELMM imposes a spatial regularization

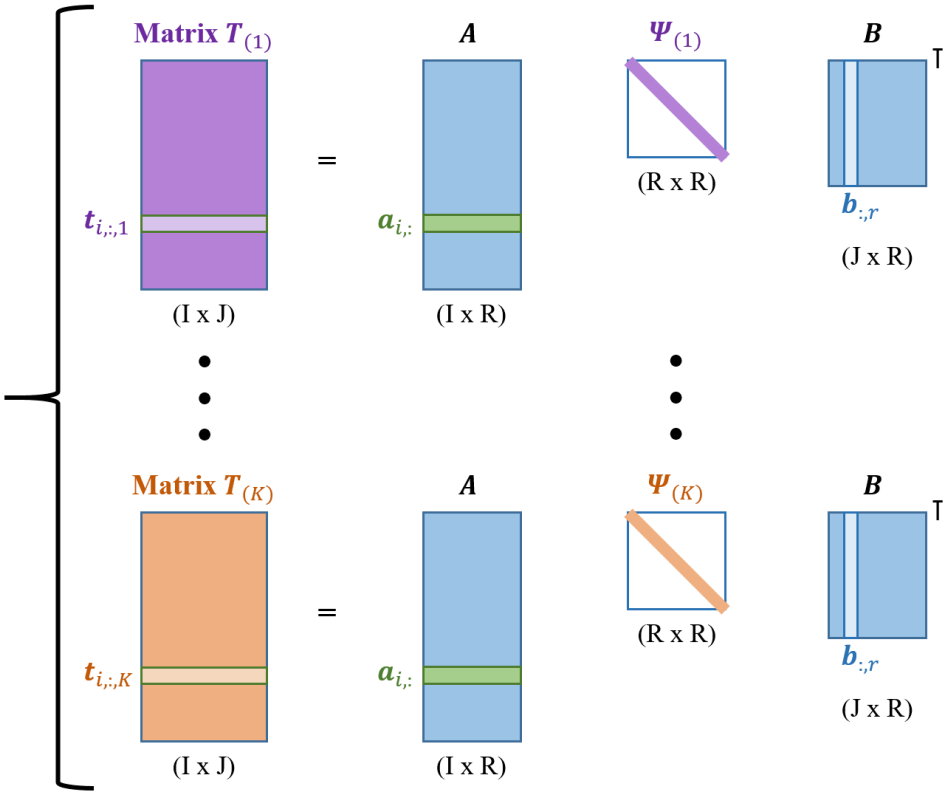


Figure 5.7: Visualization of equation (5.4) following the color scheme of figure 5.3. $\boldsymbol{\Psi}_{(k)}$ is the diagonal matrix formed from $\boldsymbol{c}_{k,:}$. The color code follows that of figure 5.8a.

on the abundances which can be seen through the cost function of NCPD, which would look like a coupled matrix decomposition (with respect to the frontal slices) having $\boldsymbol{A}$ as common factor and $\boldsymbol{B}$ subjected to scaling variabilities. Now, we point out the following:

- In RegELMM, there are as many variability functions (and convex hulls) as the frontal slices of the tensor, which is significantly lower than the number of pixels ($K \lll I$).
- In RegELMM, one row of R scaling factors in $\boldsymbol{C}$ corresponds to a full frontal slice and is shared by all the pixel rows of $\boldsymbol{A}$, while in classical ELMM, each row of scaling factors in $\boldsymbol{\Psi}$ corresponds to one pixel of $\boldsymbol{M}$ and interacts with only one row of $\boldsymbol{A}$. This clearly appears when we compare between the expressions (5.8) and (5.4) and figures 5.6 and 5.7.
- In a graphical representation, and supposing that ASC is imposed, RegELMM suggests that each convex hull contains I pixels as roughly illustrated in figure 5.8a for three

spectral signatures $\{\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3\}$ and two frontal slices, where we can also see that the relative coordinates of the pixels inside each convex hull are the same since each row of $\boldsymbol{C}$ interacts with all the elements of $\boldsymbol{A}$.

- As the frontal slices have physical tags related to how the third diversity is defined, the scaling factors of $\boldsymbol{C}$ reflect the corresponding changes that are present across the third mode on the extracted abundance and spectral components.

Regarding the last point, in other words, the *groups of pixels* or *the features of the scene* that are *highlighted or targeted* by each acquisition (e.g. time) or transformation (e.g. patches, MM) are reflected by the scaling factors in $\boldsymbol{C}$. For example, if we suppose that $\boldsymbol{T}_{(k)}$ is a transformation of $\boldsymbol{M}$ that highlights areas with relatively low brightness, $\boldsymbol{c}_{k,:}$ is expected to model the corresponding variabilities with relatively low scaling factors, which is roughly illustrated in figures 5.8b and 5.8c in the cases of patches and MM respectively with four column-components $\{\boldsymbol{b}_1/\boldsymbol{c}_1, \ldots, \boldsymbol{b}_4/\boldsymbol{c}_4\}$ and two frontal slices. In the following, we talk about these two examples.

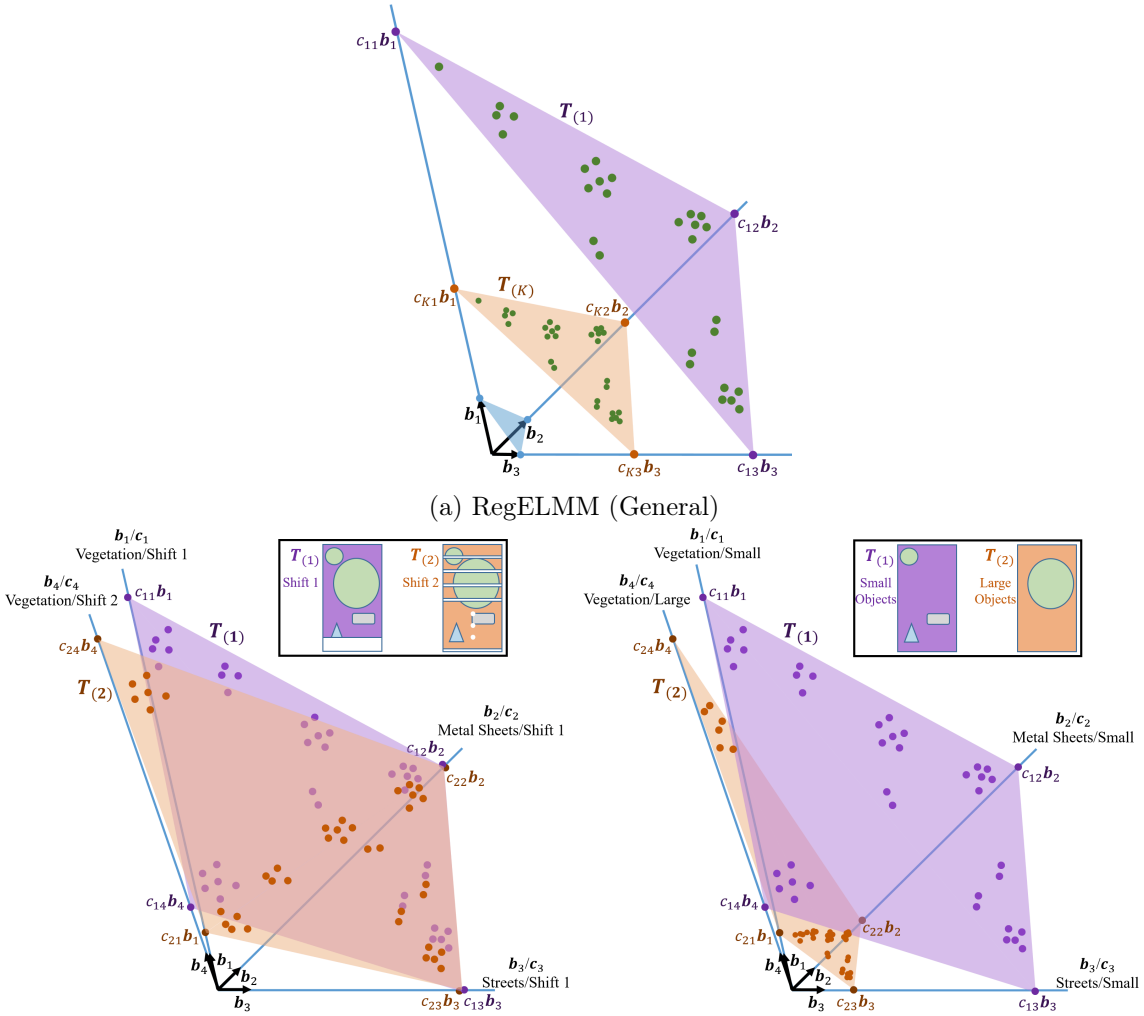
5.2.2.1 Patches

As described earlier, the frontal slices of patch-HSI tensors are essentially shifted versions of the original HSI, with $\boldsymbol{T}_{(1)}$ representing the 0-shift. Regarding the spatial regularization aspect of RegELMM, when R of NCPD is as low as the value used for NMF, this type of shifting imposes a patch-local smoothing on the abundances. Intuitively speaking, compared to NMF, it looks as if the pixel's matrix-representation in NCPD is aware of its local neighborhood.

The size of the patches are relatively very small with a radius of 1 to 3 pixels. Considering the frontal slices separately, 1- or 2-pixel shifts are almost negligible for big real HSI, so the highlighted information contained across the frontal slices is almost the same. Looking at figure 5.7, we expect that the set of $\boldsymbol{\Psi}_{(k)}$ are almost similar given that the rank of such a tensor is very low. Correspondingly, in a graphical representation, we expect the convex hulls to almost coincide.

Now, if the value of R increases, the additional abundance maps are expected to represent spatially shifted versions of the fundamental materials since the third-diversity is based on "shifting" information, which does not provide any new distinctive characteristics to the objects of the scene. This point is roughly illustrated in figure 5.8b where we have three materials: Vegetation, Metal Sheets, and Streets, but NCPD is carried out with $R = 4$. Here, we see that the convex hull of $\boldsymbol{T}_{(1)}$ gives a high scaling factor at $\boldsymbol{b}_1$ and a low one at $\boldsymbol{b}_4$, while that of $\boldsymbol{T}_{(2)}$ gives the opposite but with almost the same quantity, but both convex hulls give the same scaling factors at $\boldsymbol{b}_2$ and $\boldsymbol{b}_3$. This is due to the fact that the materials are present with almost the same quantity in both frontal slices.

Finally, patch-HSI tensors are good candidates to impose local smoothing in HU applications, but they cannot be used for multi-feature unmixing. As such, $\boldsymbol{C}$ is just a matrix of scaling factors with a low row-rank representation.



(a) RegELMM (General)

(b) RegELMM (Patches)

(c) RegELMM (MM)

Figure 5.8: Graphical representations of (a) general RegELMM in the case of three spectral signatures and two sample frontal slices, and (b) patch-RegELMM and (c) MM-RegELMM in the case of four spectral signatures (two of which have different third-diversity patterns) and two sample frontal slices. The relative coordinates of the pixels in the convex hulls (i.e. $\boldsymbol{A}$) must be the same.

5.2.2.2 Mathematical Morphology

Forming a MM-HSI tensor is explained in details in Section 4.2. In general, the frontal slices are created from successive morphological filtering of the original HSI, with each slice associated with a physical tag in terms of brightness and scale parameters, so, in addition to the spectral features, each pixel shows distinctive spatial ones. From a spatial regularization point of view, it can be seen as a smoothing within a group of shared (predefined) morphological parameters, which gives the pixels awareness of their surrounding brightness and scale characteristics. In the experiments of this chapter, we adopt the tensor-based EMP representation of Chapter 4

where the two modes of pixels are rearranged as one. Then we have a third-order tensor of dimensions $I \times J \times K$.

Looking at figure 5.7, since each image transformation in MM is seen as a characteristic of brightness and scale, we expect that the set of $\boldsymbol{\Psi}_{(k)}$ reflect the changes of brightness and scale levels of the extracted materials per frontal slice. Moreover, if the value of R increases, NCPD is expected to show abundances with different morphological properties as well as the spectral ones. This point is roughly illustrated in figure 5.8c. Here, we see that the convex hull of $\boldsymbol{T}_{(1)}$ gives high scaling factors at $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $\boldsymbol{b}_3$, which correspond to small scales (supposedly highlighted by $\boldsymbol{T}_{(1)}$), and a low scaling factor at $\boldsymbol{b}_4$, which corresponds to large scales. On the other hand, $\boldsymbol{T}_{(2)}$ shows the opposite since it supposedly highlights large features.

As such, in the case where the transformations hold physical significance, so does RegELMM, which jointly benefits from a spatial regularization and a multi-feature unmixing with modeled variabilities. Consequently, $\boldsymbol{C}$ holds a two-way significance: (a) scaling factors in the rows, and (b) morphological patterns in the columns.

Finally, we note that one has to be careful when deciding the number of transformations (i.e. the value K) and their parameters, which should be enough for the tensor to have meaningful features. For instance, a very small value of K may lead to poor spatial diversity, and a very high value of K may lead to spatial redundancy and rank issues.

5.3 Abundance Sum-to-one Constraints in CPD

In this section, we consider that the elements of NCPD including AO-ADMM and the discussions on uniqueness have already been introduced in the earlier chapters, especially in Chapter 2. As such, we suffice by talking about how we implement ASC constraints in AO-ADMM for third-order tensors.

For the modeling and computation, in order to model the CPD problem as close as possible to LMM, we adopt the ASC constraints. The key point is to extend the concept from NMF to NCPD. In the following, items 1 through 4 explain the initialization of $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{I \times J \times K}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times R}$, and item 5 denotes their updates *after each AO-ADMM iteration* (i.e. after the three ADMM sub-problem updates):

1. ASC means that: $\sum_{r=1}^{R} a_{ir} = 1 \; \forall \; i \in \{1, \ldots, I\}$.

2. Knowing that $t_{i,j,k} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$, and assuming that $\lambda_{rrr} = 1 \; \forall \; r$, in order to arrive at $\sum_{r=1}^{R} a_{ir} = 1$, we start by introducing a vertical column in $\boldsymbol{\mathcal{T}}$ such that $t_{i,J+1,K} = \delta$ $\forall \; i \in \{1, \ldots, I\}$ (where $\delta > 0$).

3. Correspondingly, we introduce a row vector in $\boldsymbol{B}$ such that $b_{J+1,r} = \delta c_{K,r}^{-1} \; \forall \; r \in \{1, \ldots, R\}$. Again, knowing that $t_{i,j,k} = \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}$, this verifies that $t_{i,J+1,K} = \sum_{r=1}^{R} a_{ir} \delta c_{K,r}^{-1} c_{Kr} = \delta$.

4. However, the row vector $\boldsymbol{b}_{J+1,:}$ in $\boldsymbol{B}$ also corresponds to a whole slice in $\boldsymbol{\mathcal{T}}$, i.e., $\boldsymbol{\mathcal{T}}_{:,J+1,:}$. Then $\forall\ k \in \{1,\ldots,K-1\}$, $\boldsymbol{\mathcal{T}}$ should be updated following $t_{i,J+1,k} = \delta \sum_{r=1}^{R} a_{ir}(c_{kr}/c_{K,r})$.

5. Finally, we introduce the per-iteration updates of $\boldsymbol{\mathcal{T}}$ and $\boldsymbol{B}$, denoted by $\tilde{\boldsymbol{\mathcal{T}}}$ and $\tilde{\boldsymbol{B}}$, which are applied after each AO-ADMM iteration:

$$\tilde{\boldsymbol{\mathcal{T}}} = \left[\boldsymbol{\mathcal{T}}_{(k)} \ \middle|\ \delta \left[\sum_{r=1}^{R} a_{ir}(c_{kr}/c_{K,r})\right]_{\forall\, i\in\{1,\ldots,I\}}\right]_{\forall\, k\in\{1,\ldots,K\}},$$

meaning that each frontal slice in $\boldsymbol{\mathcal{T}}$ has an additional column that is updated following $\delta \sum_{r=1}^{R} a_{ir}(c_{kr}/c_{K,r})\ \forall\ i \in \{1,\ldots,I\}$.

$$\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} \\ \delta \left[c_{K,r}^{-1}\right]_{\forall\, r\in\{1,\ldots,R\}} \end{bmatrix},\ \delta > 0$$

meaning that $\boldsymbol{B}$ has an additional row that should be updated following $\delta \left[c_{K,r}^{-1}\right]\ \forall\ r \in \{1,\ldots,R\}$. The updates of $\tilde{\boldsymbol{\mathcal{T}}}$ and $\tilde{\boldsymbol{B}}$ theoretically ensure that the next AO-ADMM iteration takes into account that $\sum_{r=1}^{R} a_{ir} = 1\ \forall\ i \in \{1,\ldots,I\}$.

5.4 Experiments and Results

In this section, we talk about the experiments that were carried out and discuss the results. First, we start with a description of the HSI datasets that were used. Then, we show some differences between NMF and NCPD over a synthetic HSI example. After that, we present and discuss the results of each of the real HSI separately. For each real HSI, the results are presented under two main categories:

- A comparison of HU between NMF and NCPD, which focuses on the multi-feature unmixing aspect by showing qualitative and quantitative comparisons of the extracted features as well as the use of constraints.
- A analysis of ELMM, which focuses on the spatial regularization and spectral variability aspects by showing a comparative analysis between patch- and MM-HSI NCPD done on real data based on Section 5.2.

In each experiment, among 20 random initializations of the factor matrices, the result with the minimum root mean squared error (RMSE) is chosen. The estimated spectral signatures of $\boldsymbol{B}$ are identified based on their minimum spectral angular distance (SAD), in degrees, with respect to the reference spectra of the HSI:

$$SAD(\vec{\boldsymbol{e}}, \vec{\boldsymbol{b}}) = \arccos\left(\frac{\vec{\boldsymbol{e}} \cdot \vec{\boldsymbol{b}}}{\|\vec{\boldsymbol{e}}\| \times \|\vec{\boldsymbol{b}}\|}\right) \tag{5.9}$$

The maps and plots that are presented here as the results of the experiments represent the *columns* of the factor matrices. In all of the following, the term *NMF* refers to Sparse-NMF (i.e. sparse over $\boldsymbol{A}$), while *NCPD* refers to CPD with only nonnegativity imposed.

Finally, we note that quantitative validation for abundance maps is usually not evident, so we rather present a qualitative comparison in this case. In a future submission of this work, particularly with the urban dataset where ground-truth fractional abundance maps are provided, we plan a quantitative comparison on the estimated fractions noting, however, that it is still not clear how to quantitatively compare the abundance maps between matrix-based ground-truth and tensor-based results since the latter are usually more specific in terms of unmixing the components (as shown later in this section).

5.4.1 Description of Data Sets

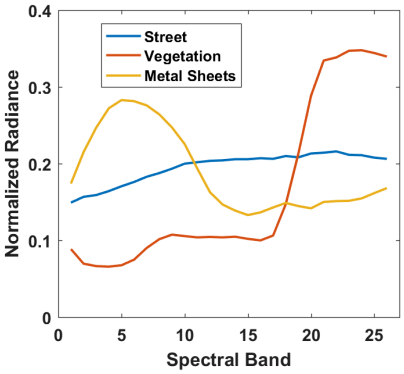
5.4.1.1 Synthetic HSI

The synthetic HSI (figure 5.9a in false colors) has dimensions 128×128 and contains six objects that vary in size (figure 5.9d). Three spectral signatures, {*Street*, *Vegetation*, *Metal Sheets*}, were taken from the HSI of Pavia University (by averaging their reference pixels) and downsampled to 26 bands (figure 5.9b). Each object was assigned a mixture of the three signatures (figure 5.9c) such that every two signatures are close-looking. Then, the rank of the matricized HSI is 3. As for the MM-HSI tensor, morphological filtering was done using Openings by reconstruction with successive sizes of the SE, and the corresponding frontal slices are shown in figure 5.9e in grayscale. Then, the rank of the tensor is 6.

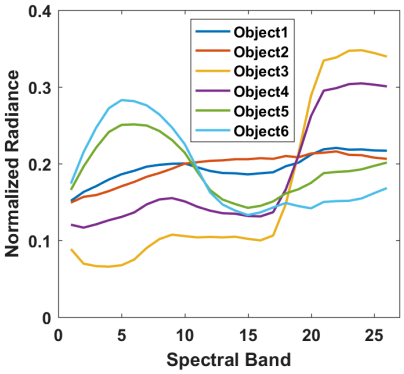
5.4.1.2 Pavia University

This HSI was acquired by the ROSIS sensor over the University of Pavia in Italy. Its spatial dimensions are 610×340 with a geometric resolution of 1.3 meters, and it consists of 103 spectral bands. The groundtruth (GT) image is included in the data set and it consists of nine classes (figure 5.10b). We use the pixels of the GT classes to extract a spectral reference of the nine classes (as endmembers) by averaging their spectral responses for each class. Figures 5.10a, 5.10b, and 5.10c show the HSI in false colors along with its spatial GT and extracted spectral reference respectively. In figure 5.10c, some classes have very similar spectral signatures (e.g. *Asphalt*, *Gravel*, *Bitumen*, and *Bricks*). Such classes are sometimes confused when the extracted signatures are compared to the reference, so they are treated as one especially that unmixing is blind.

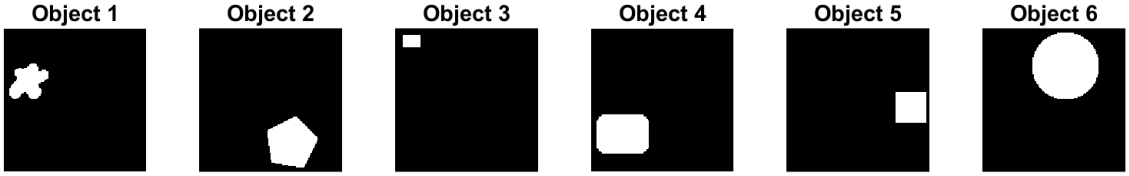
(a) False colors



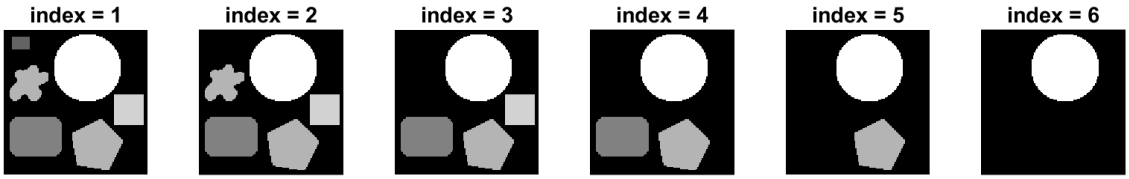
(b) Endmembers



(c) Spectra



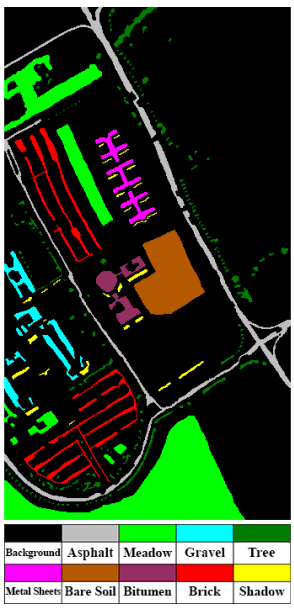
(d) The different objects of the image



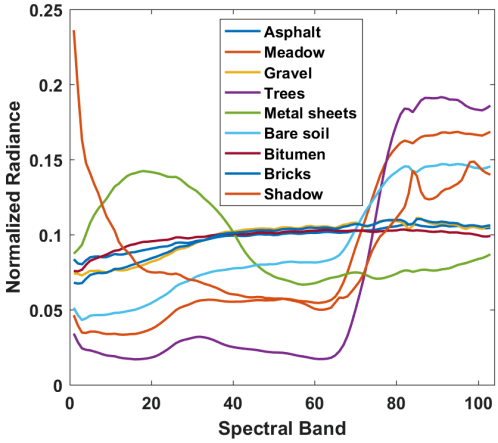
(e) Morphological filtering of the image

Figure 5.9: The different elements of the synthetic HSI tensor

(a) Pavia. False colors



(b) Pavia. Spatial GT



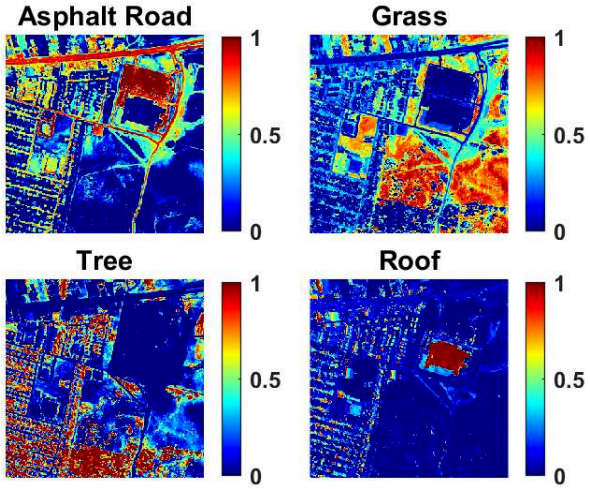
(c) Pavia

Figure 5.10: Pavia HSI in false colors and its spatial groundtruth (GT)

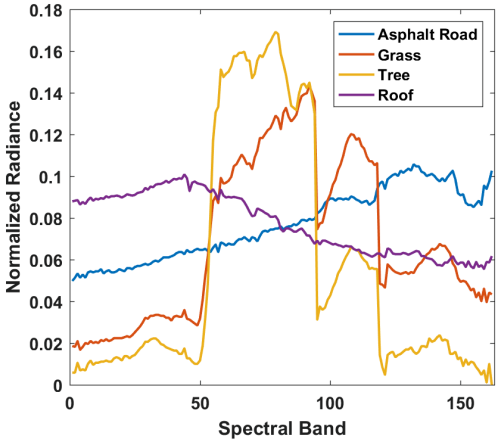
5.4.1.3 Urban

This HSI was acquired over an urban area at Copperas Cove, US by the HYDICE sensor. Its spatial dimensions are 307×307 pixels with a geometric resolution of 2 meters, and it consists of 162 spectral bands. The GT image is included in the data set and it consists of four endmembers used as a spectral reference (figure 5.11b). Figures 5.11a, 5.11b, and 5.11c show the HSI in false colors along with its abundance GT and spectral reference respectively.

(a) Urban. False colors



(b) Urban. Abundance GT



(c) Urban

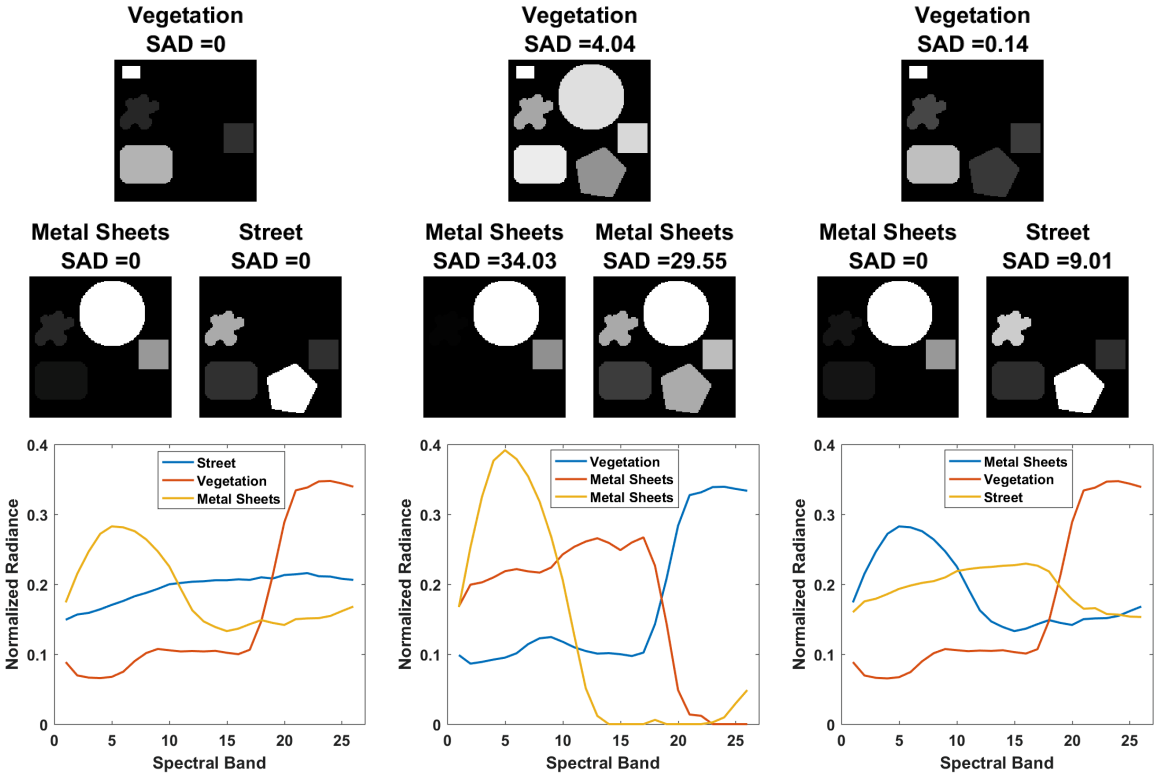
Figure 5.11: Urban HSI in false colors and its abundance GT. Spectral references of Pavia and Urban HSI

5.4.2 Results - Synthetic Data

Here, we compare the results of unmixing a synthetic HSI between NMF and NCPD as simple examples on the previous discussions. We note that (a) the ideal ranks of NMF and NCPD are $R = 3$ and $R = 6$ respectively, (b) the GT tensor is built such that Kruskal's condition

(2.19) is verified, and (c) in the case of NCPD, figures 5.9d and 5.9c are seen as the abundance and spectral GT of the tensor respectively.

Looking at figure 5.12, we see that NMF unmixes the data based only on spectral features, while looking at figure 5.13, we see that NCPD does it based jointly on spectral and spatial features. Moreover, we notice through figures 5.12b and 5.12c that NMF does not reach its GT (figure 5.12a) without constraints, while we see through figures 5.13a and 5.13b that NCPD perfectly recovers the GT components in $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively.



(a) GT components (b) NMF (no ASC) (c) NMF (ASC)

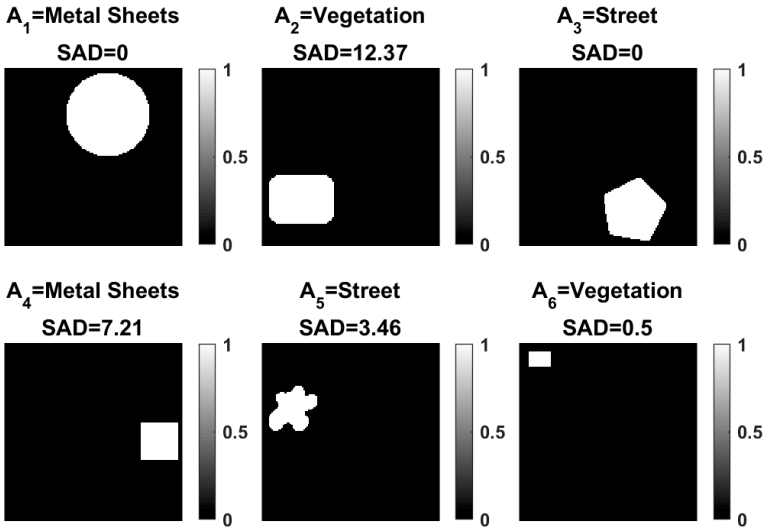
Figure 5.12: The GT and NMF results of the matricized synthetic HSI

As for $\boldsymbol{C}$, the morphological patterns shown in figure 5.13c reflect how each object was filtered across the frontal slices as shown in figure 5.9e. In general, the quantity of the objects' existence in each of the slices is the same, which is reflected by plots that are horizontal across the morpho-indices where the objects remain unfiltered, and drop to 0 where the objects are filtered out. For example, $\boldsymbol{A}_1$ represents Object 6 (the circle), which remains unfiltered across all the slices, which is reflected by a constant plot of $\boldsymbol{C}_1$. On the other hand, $\boldsymbol{A}_4$ is filtered out after the third frontal slice, which is reflected by a drop of $\boldsymbol{C}_4$ to 0 after the third morpho-index.

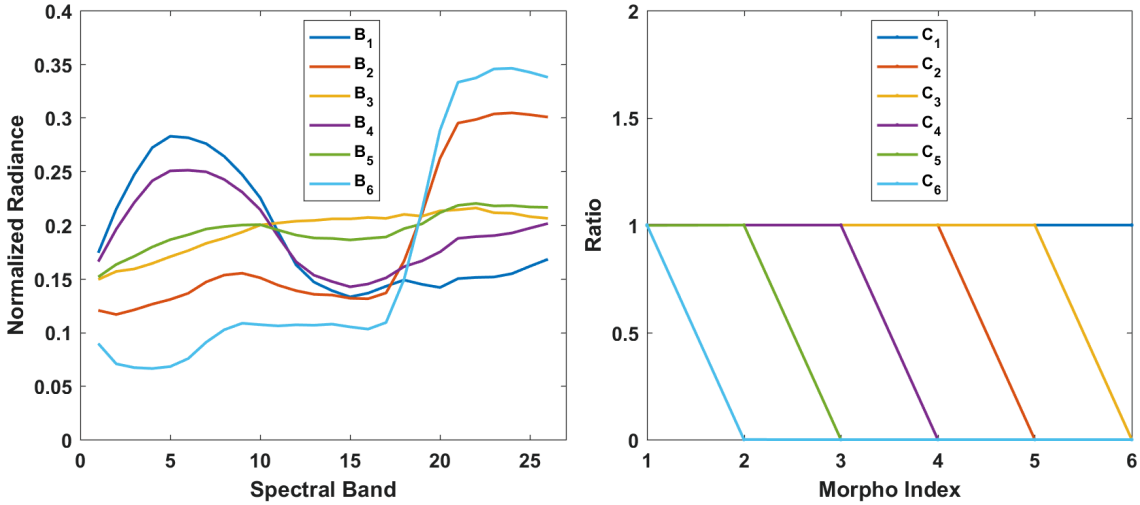
Now, we give some notes regarding the SAD values obtained by NCPD and the identification of the components. Objects 2, 3 and 6 are assigned pure elements, which is respectively reflected by 0 SAD values in the columns $\{\boldsymbol{B}_3, \boldsymbol{B}_6, \boldsymbol{B}_1\}$. On the other hand, Objects 1, 4 and

5 are assigned mixtures with a majority of 80% *Street*, 70% *Vegetation* and 60% *Metal Sheets* respectively, and since the unmixing respectively separated them in the individual columns $\{\boldsymbol{B}_5, \boldsymbol{B}_2, \boldsymbol{B}_4\}$, their endmember fractional abundances are not identified, but their major compositions are. Hence, unlike NMF, NCPD does not necessarily observe fractional sum-to-one abundances from a pure spectral perspective. The interpretation of the decomposition and the identification of components become different and sometimes more specific depending on the third-mode diversity.

Finally, this was indeed a simple and easy example in order to build an intuition for more complicated structures, which is the case for real HSI that we present next.



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$ (c) Components of $\boldsymbol{C}$

Figure 5.13: NCPD results (without constraints) of the synthetic MM-HSI tensor

5.4.3 Results - Pavia University

In this section, we present the experiments carried out on the HSI of Pavia using NMF and NCPD. We start by explaining and comparing the results of NMF and NCPD. After that, we focus on the RegELMM analysis and properties by comparing the cases of patch-HSI and MM-HSI tensors. The MM-HSI tensor is built using EMP with Openings and Closings by reconstruction as explained in Section 5.2.2.2. The structuring elements are disks with the successive radii (in pixels): $\{2, 7, 12, 17\}$, which corresponds to $K = 9$ frontal slices shown in figure 5.14 in false colors from Closings to Openings by reconstruction. The patch-HSI tensor is built using 3×3 patches, which corresponds to $K = 9$ frontal slices shown in figure 5.15. In both cases, after reordering the pixels in lexicographic order, the data tensor has the dimensions $207400 \times 103 \times 9$. As for the rank, we find that $R = 4$ and $R = 8$ are the best values for NMF and NCPD respectively.

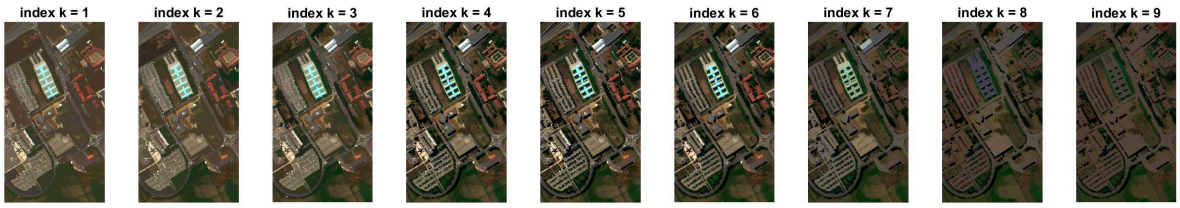


Figure 5.14: False colors of the transformations that form the EMP of the MM-HSI tensor of Pavia. The differences across the slices are clearly noticeable. This supports the physical significance of MM for unmixing in RegELMM (i.e., if we project these slices on figure 5.7, we can tell how the variability function through $\boldsymbol{\Psi}_{(k)}$ will be affected by the morphological characteristics).

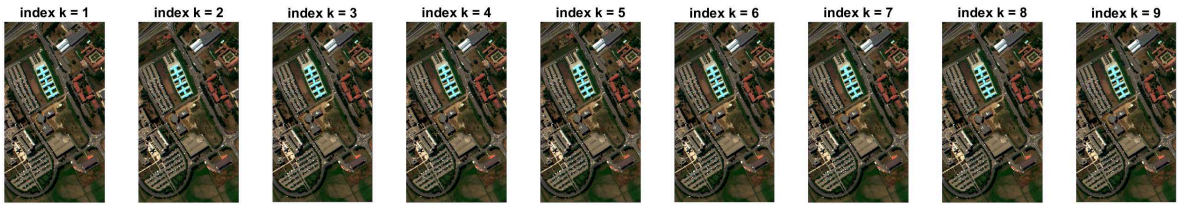
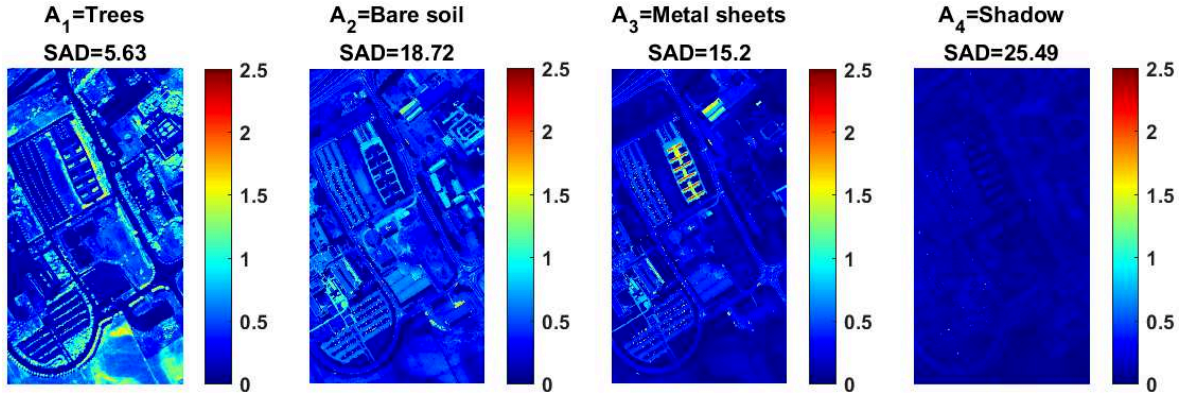


Figure 5.15: False colors of the slices that form the 3×3 patch-HSI tensor of Pavia. The shifts across the slices are not noticeable, so they look visually the same. This is why patches have no physical significance for unmixing in RegELMM (i.e., if we project these slices on figure 5.7, we can tell how the variability function through $\boldsymbol{\Psi}_{(k)}$ represents almost the same information for each slice).

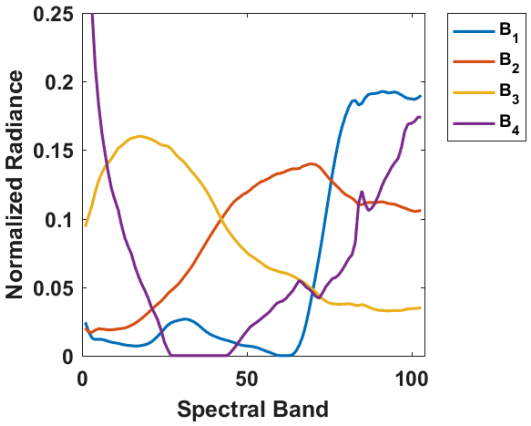
5.4.3.1 NMF Results

Figure 5.16 shows the abundance maps and the spectral signatures obtained in $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively. Above each abundance map, we see the class that was assigned to it based on the minimum SAD value that is shown as well.

First, we look at the first three components, *Trees*, *Bare Soil*, and *Metal Sheets*. Their spectral signatures that appear in figure 5.16b look very similar to those of the reference except for that of *Bare Soil* ($\boldsymbol{B}_2$), which is slightly different, and the corresponding SAD values are relatively low. Looking at the abundance maps, we see that $\boldsymbol{A}_1$ focuses on some tree and vegetation areas, $\boldsymbol{A}_2$ focuses on some soil and bricks areas, and $\boldsymbol{A}_3$ focuses on metal sheets areas. However, we notice that other areas of the scene do not or barely appear (asphalt road, brick parking lots, other soil areas, etc) due to the insufficiency of LMM to model their variabilities, which manifests fully in the *Shadow* component. Regarding the latter, the spectral signature looks slightly similar to the reference, but the shadows in the map are barely visible due to their relatively very low brightness.



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$

Figure 5.16: Pavia. NMF results (with ASC) of the matricized HSI for $R = 4$

5.4.3.2 NCPD Results

Under different constraints, figures 5.17 and 5.18 show the abundance maps, spectral signatures, and morphological patterns obtained in $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ respectively. Regarding the figures and plots, we note that matrix and tensor decomposition models suffer from an unavoidable

scaling ambiguity of the components such that $\forall \alpha_r \in \mathbb{R}^*$:

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} (\alpha_r \alpha_r^{-1}) \boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r = \sum_{r=1}^{R} \boldsymbol{a}_r \otimes (\alpha_r \boldsymbol{b}_r) \otimes (\alpha_r^{-1} \boldsymbol{c}_r)$$

which means that the columns of the matrices may not be scaled correctly, and more specifically, that the *scaling factors* of $\boldsymbol{C}$ may not represent the true ones. As such, we choose to show the normalized versions of the columns of $\boldsymbol{B}$ for a better visual comparison with respect to the spectral reference, but we show those of $\boldsymbol{C}$ without modification.

First, we start with figures 5.17 and 5.18, both of which represent NCPD with ASC, but the former does not impose sparsity while the latter does. Between the two figures, the components look almost the same. In the following, we look into each of the factor matrices separately (from the perspective of figure 5.17). Seeing as the components of the figures are not sorted in the same order, we make the following correspondences from the perspective of $\boldsymbol{A}$, which makes it easier to associate the differences:

Figure 5.17	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_3$	$\boldsymbol{A}_4$	$\boldsymbol{A}_5$	$\boldsymbol{A}_6$	$\boldsymbol{A}_7$	$\boldsymbol{A}_8$
Figure 5.18	$\boldsymbol{A}_3$	$\boldsymbol{A}_1$	$\boldsymbol{A}_2$	$\boldsymbol{A}_5$	$\boldsymbol{A}_4$	$\boldsymbol{A}_6$	$\boldsymbol{A}_7$	$\boldsymbol{A}_8$

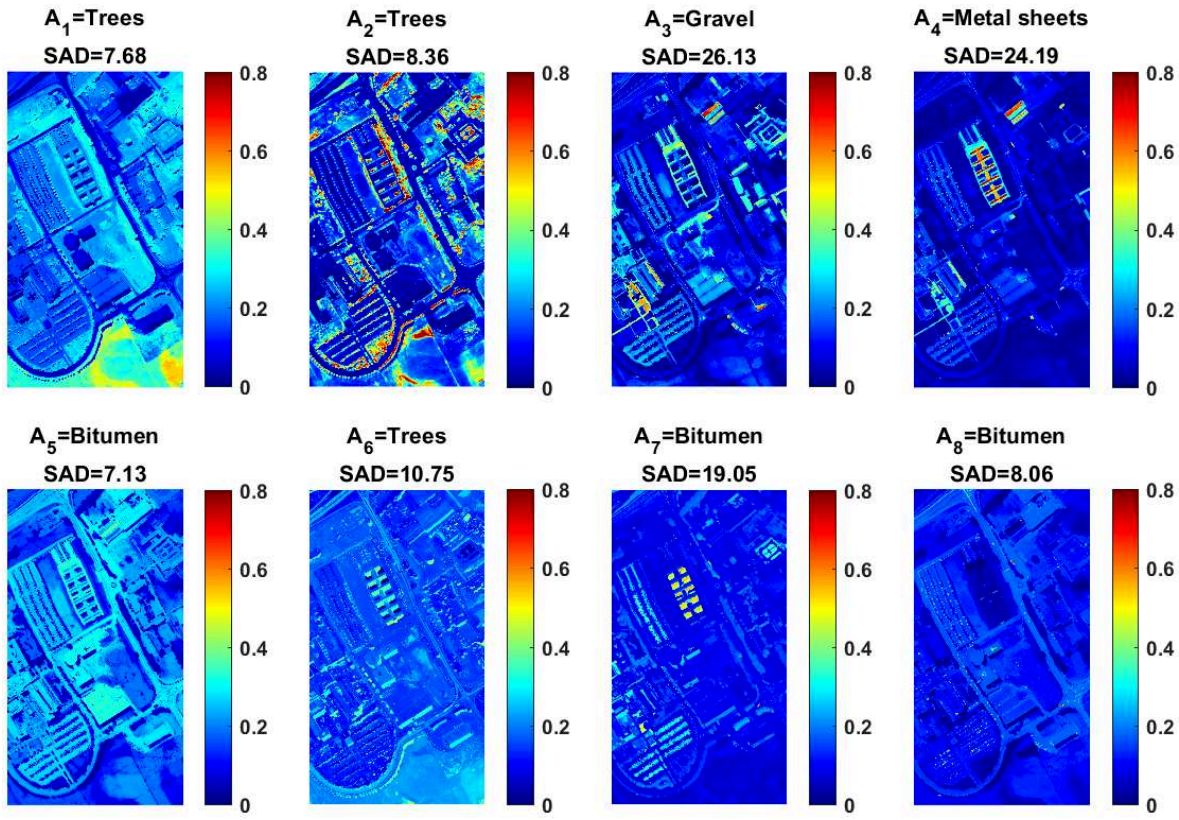
NCPD Abundance Maps: In general, we notice through figure 5.17a that the decomposition is done based jointly on spectral and morphological properties with fairly good results. The components can be put into three groups: (a) $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_6\}$, (b) $\{\boldsymbol{A}_3, \boldsymbol{A}_5, \boldsymbol{A}_7, \boldsymbol{A}_8\}$, and (c) $\{\boldsymbol{A}_4\}$.

$\boldsymbol{A}_1$, $\boldsymbol{A}_2$, and $\boldsymbol{A}_6$ were identified as *Trees*. $\boldsymbol{A}_1$ highlights big vegetation areas like meadow, $\boldsymbol{A}_2$ highlights small vegetation areas like trees, and $\boldsymbol{A}_6$ highlights vegetation with low illumination like shadows and some bare soil and meadow areas. Here, we note that the reference spectra of *Trees* and *Meadow* look similar, and that that of *Bare Soil* comes between them and the group of *Bitumen*-like spectra. As such, we notice that areas of bare soil can appear in the abundance maps that belong to both groups of vegetation and bitumen.

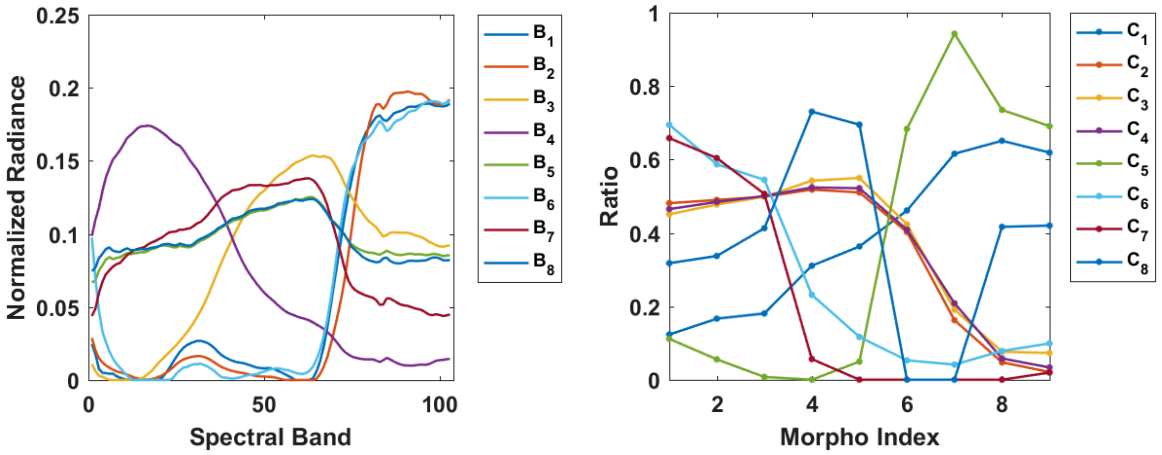
$\boldsymbol{A}_3$, $\boldsymbol{A}_5$, $\boldsymbol{A}_7$, and $\boldsymbol{A}_8$ were identified as *Gravel* and *Bitumen*. $\boldsymbol{A}_3$ focuses on small-to-medium bare soil and gravel areas and roads, $\boldsymbol{A}_5$ highlights big connected areas of artificially constructed objects such as roads (asphalt), building roofs (bitumen, bricks and gravel), and parking lots (bricks), $\boldsymbol{A}_7$ highlights some relatively dark parking lots and building shadows, and $\boldsymbol{A}_8$ shows tiny bright dots which represent vehicles. As mentioned earlier, the reference spectra of *Bitumen*, *Gravel*, *Asphalt Road*, and *Bricks* look similar.

$\boldsymbol{A}_4$ was identified as *Metal Sheets*. This component looks similar to the one obtained by NMF, but with better highlighting of the components.

Now, compared to the *Trees* component obtained using NMF, we have three components, each of which represents vegetation-related features of certain scale and brightness properties and highlights them better, especially the shadows, which can be attributed to the RegELMM property.



(a) Components of $\boldsymbol{A}$



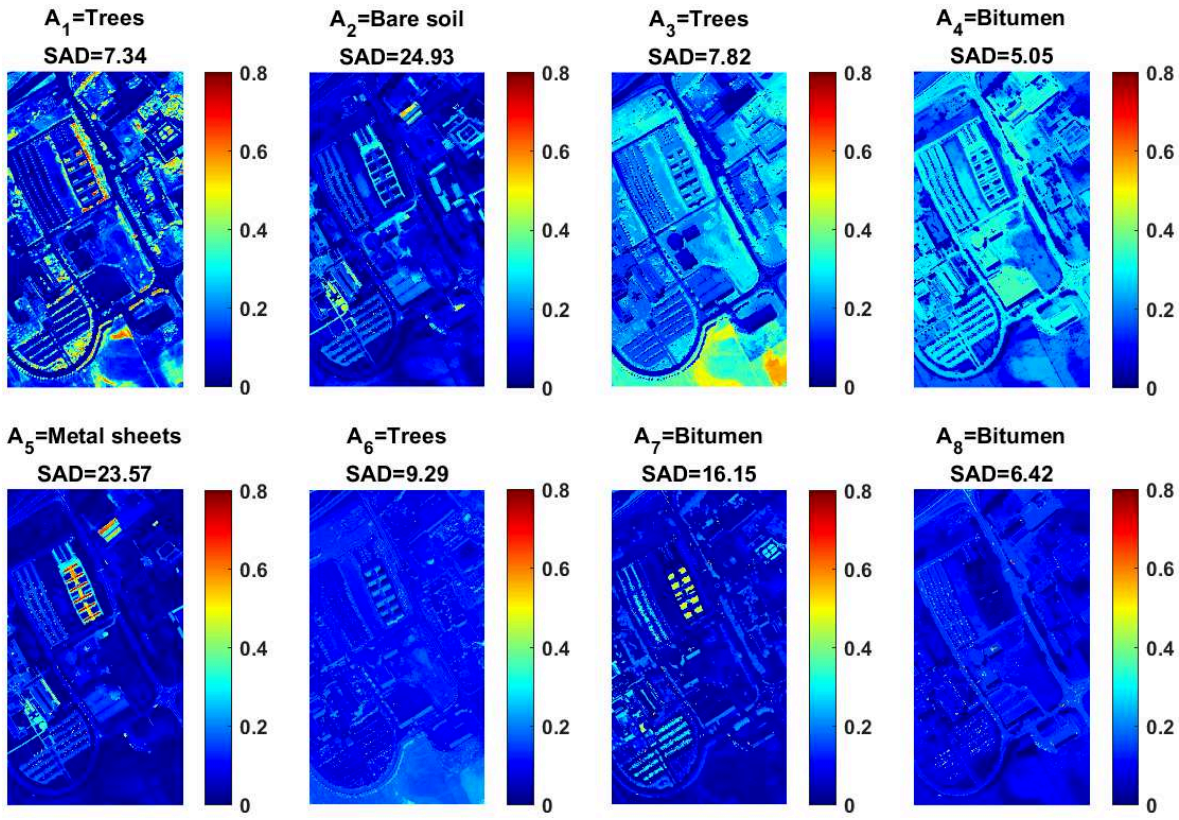
(b) Components of $\boldsymbol{B}$

(c) Components of $\boldsymbol{C}$

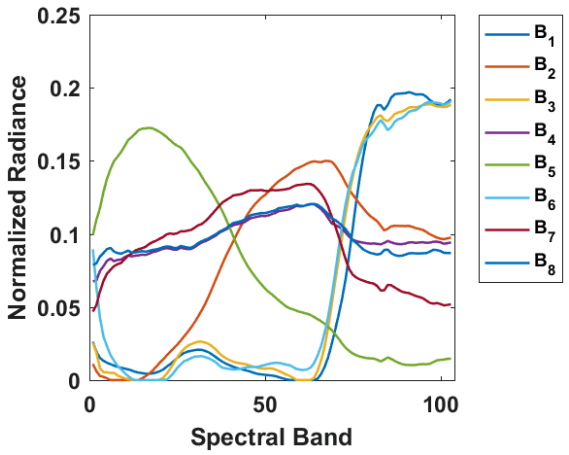
Figure 5.17: Pavia. NCPD results (with ASC) of the MM-HSI tensor for $R = 8$

Compared to the *Bitumen* component obtained from NMF, we have four components of differently highlighted spatial features that were not observed in the case of NMF like asphalt roads and shadows.

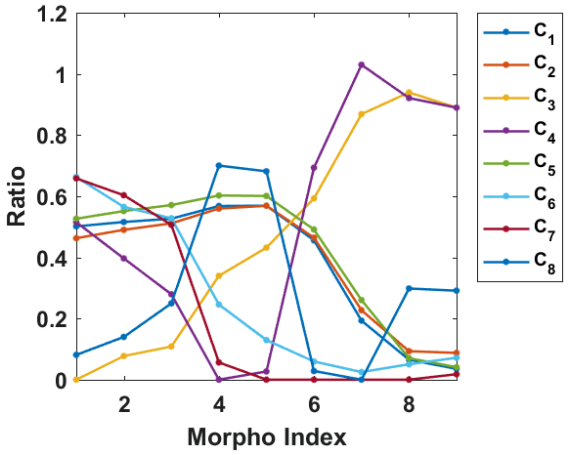
NCPD Spectral Components: Here, we are interested in the quality of the spectral sig-



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$



(c) Components of $\boldsymbol{C}$

Figure 5.18: Pavia. NCPD results (with Sparsity and ASC) of the MM-HSI tensor for $R = 8$

natures and their SAD values. In general, the plots of figures 5.17b and 5.18b reflect the features that appear in their corresponding abundance maps. Compared to the SAD values obtained using NMF, those obtained using NCPD are lower or comparable. The components are grouped similarly to those of $\boldsymbol{A}$:

- $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, and $\boldsymbol{B}_6$ are detected as vegetation with low minimal SAD values. While they look almost collinear, their corresponding components in $\boldsymbol{A}$ and $\boldsymbol{C}$ explain the results well, which applies to the other components.
- $\boldsymbol{B}_3$, $\boldsymbol{B}_5$, $\boldsymbol{B}_7$, and $\boldsymbol{B}_8$ were detected as *Bitumen* and *Gravel*. $\boldsymbol{B}_5$, $\boldsymbol{B}_7$ and $\boldsymbol{B}_8$ are very close to the reference spectra. This is not the case for $\boldsymbol{B}_3$, which is similar to *Bare Soil* obtained using NMF. The SAD values are slightly improved when sparsity is imposed.
- $\boldsymbol{B}_5$ shows the *Metal Sheets* pattern.

NCPD Morphological Components: Here, we are interested in the significance of the patterns and their relevance to the original HSI image. First, looking at the graphs of figure 5.17c and 5.17c, we observe four main patterns that can be associated to the chosen morphological parameters:

- $\boldsymbol{C}_6$ and $\boldsymbol{C}_7$ correspond to dark features like shadows, i.e. objects that have low brightness or illumination. These two curves have higher values when k (the morpho-index) corresponds to *Closings by reconstruction*, then continue decreasing towards Openings by reconstructions.
- $\boldsymbol{C}_2$, $\boldsymbol{C}_3$, and $\boldsymbol{C}_4$ correspond to small features. These curves have higher values when k corresponds to Openings and Closings by reconstruction with *small SE*.
- $\boldsymbol{C}_1$ and $\boldsymbol{C}_4$ correspond to big features. These curves have higher values when k corresponds to Openings by reconstruction with *big SE*.
- $\boldsymbol{C}_8$ corresponds to the tiny vehicles. The curve is very high when k corresponds to the Closing by reconstruction with the *smallest SE*, and to the original image.

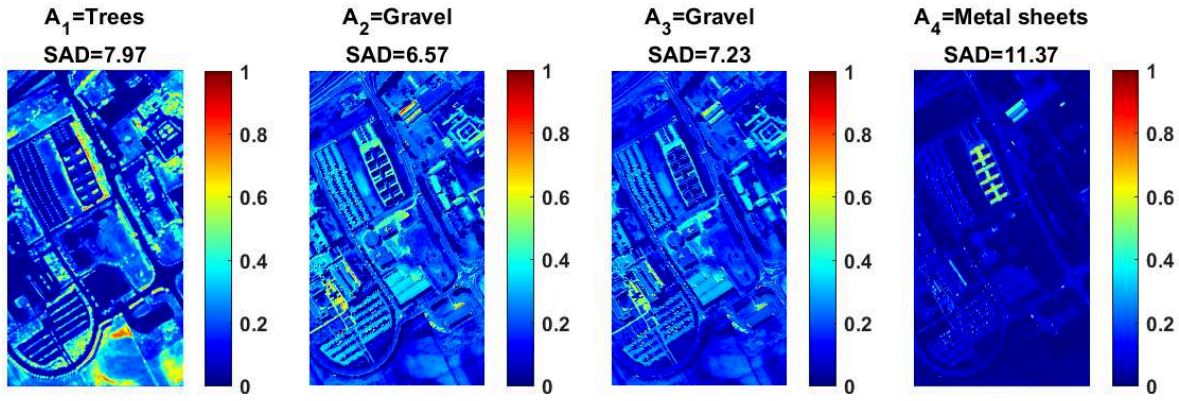
Now, knowing that the index $k = 5$ corresponds to the fifth row of $\boldsymbol{C}$, which is responsible for reconstructing the original HSI slice inside the tensor, it is interesting to see how the scaling factors are reflected in the original HSI, so we note the following by looking at figure 5.17c where $k = 5$:

- $\boldsymbol{C}_6$ and $\boldsymbol{C}_7$ have low values (dark shadowy features).
- $\boldsymbol{C}_2$, $\boldsymbol{C}_3$, $\boldsymbol{C}_4$, and $\boldsymbol{C}_8$ have high values (relatively bright features including those of vehicles).
- $\boldsymbol{C}_1$ has a slightly low value (dark asphalt roads, building roofs, parking lots, bare soil areas and meadow).

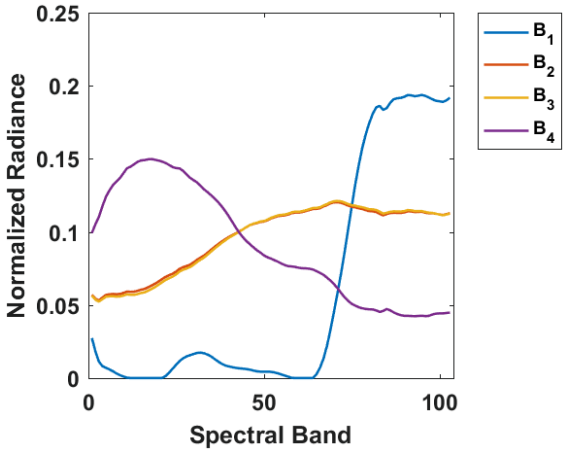
These relationships show the dual significance of $\boldsymbol{C}$ (columns and rows) and how NCPD is able to highlight some features while at the same time reconstruct the original image. They are also associated to RegELMM, which we talk about next and compare the results to patch-HSI tensors.

5.4.3.3 ELMM Analysis, Comparison Between Patches and MM tensors

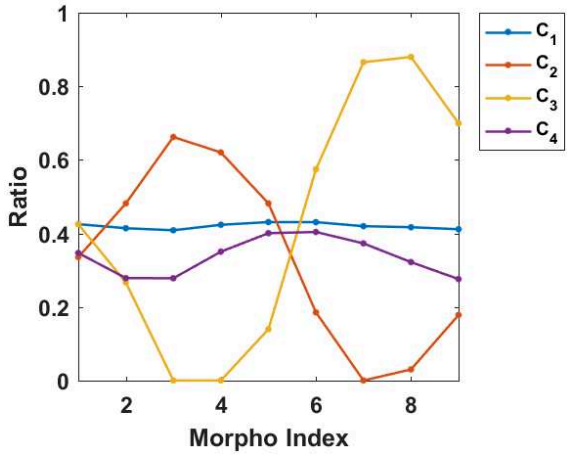
Here, we are interested in the properties of RegELMM, which can be showcased through the comparison between the cases of patches, where the model has no physical significance, and that of MM, where the model contains scale and brightness characteristics. We have two tensors based on patches and MM, and we decompose them using NCPD with ASC. Since patch-HSI tensors are expected to be decomposed using the rank of NMF, we compare the results for both tensors when $R = 4$. This demonstrates the *spatial regularization* aspect of MM when the rank is low. Additionally, in order to demonstrate how patch-NCPD shows "duplicated" components as the rank increases, we we compare the results for both tensors when $R = 8$. We compile the discussion into three stages: (a) figure 5.19 which represents the patch-HSI NCPD where $R = 4$, (b) figure 5.20 which represents the MM-HSI NCPD where $R = 4$, and (c) figure 5.21 which represents the patch-HSI NCPD where $R = 8$.



(a) Components of $\boldsymbol{A}$



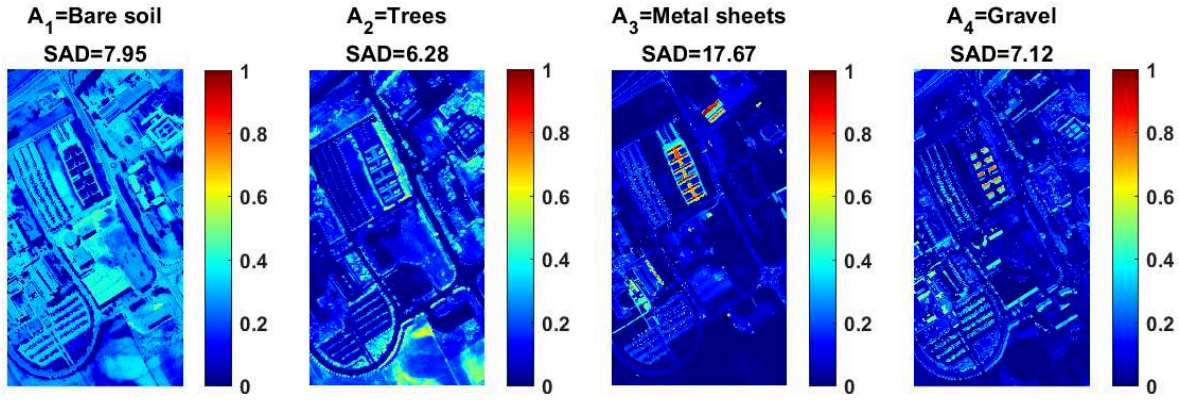
(b) Components of $\boldsymbol{B}$



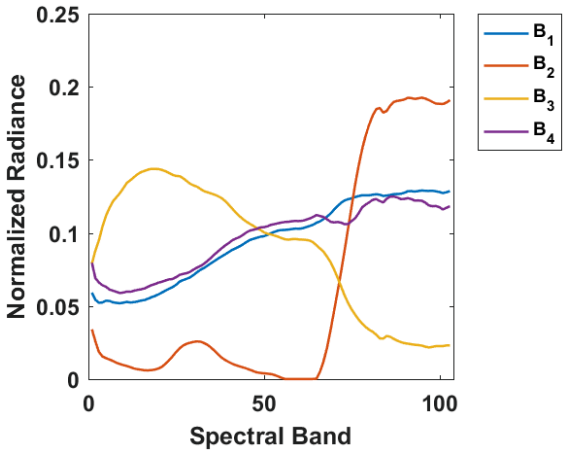
(c) Components of $\boldsymbol{C}$

Figure 5.19: Pavia. NCPD results of the Patch-HSI tensor for $R = 4$

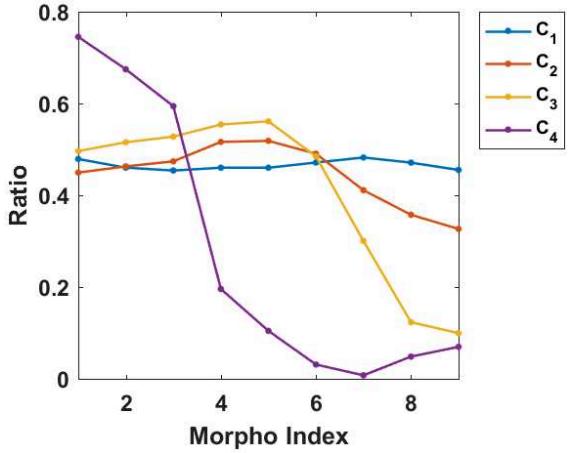
Looking at figure 5.19, we start with the abundance maps. $\boldsymbol{A}_1$ and $\boldsymbol{A}_4$ represent *Trees* and *Metal Sheets* and can be compared to those obtained by NMF in figure 5.16. As for $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$, they represent *Gravel* and can be associated to the *Bare Soil* component obtained by



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$

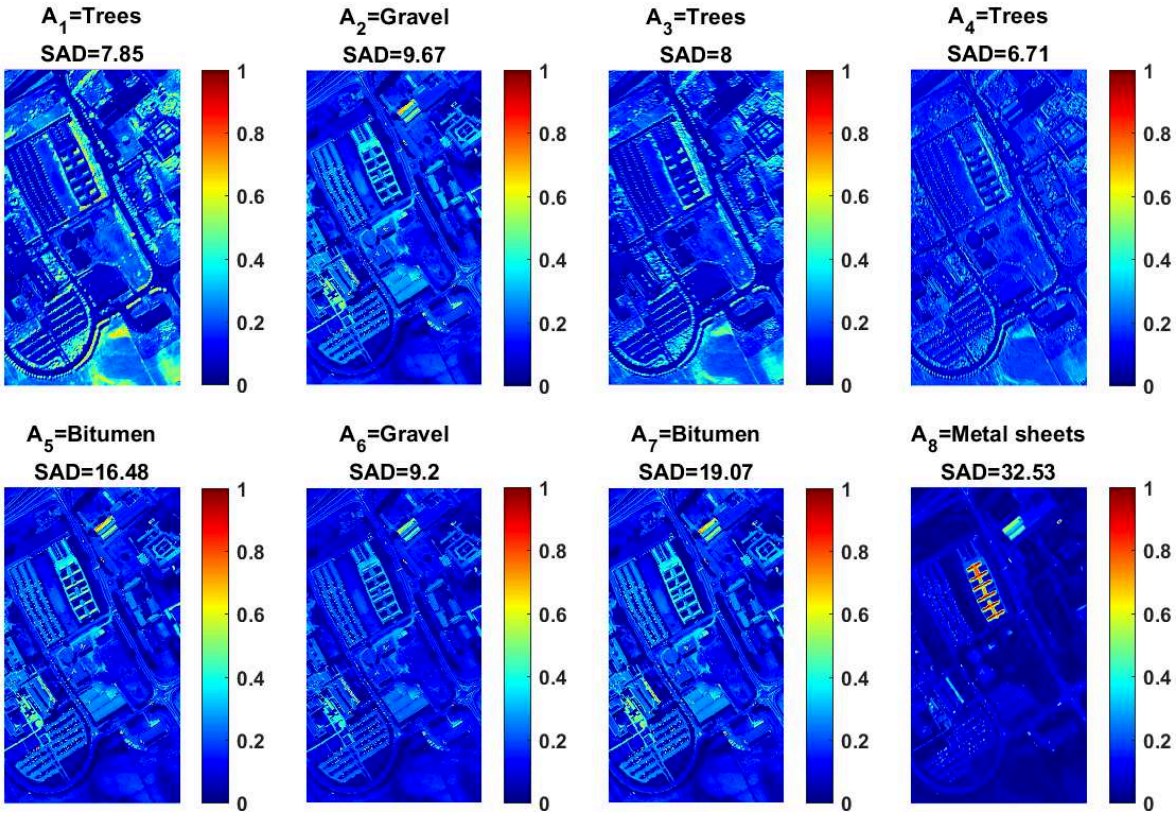


(c) Components of $\boldsymbol{C}$

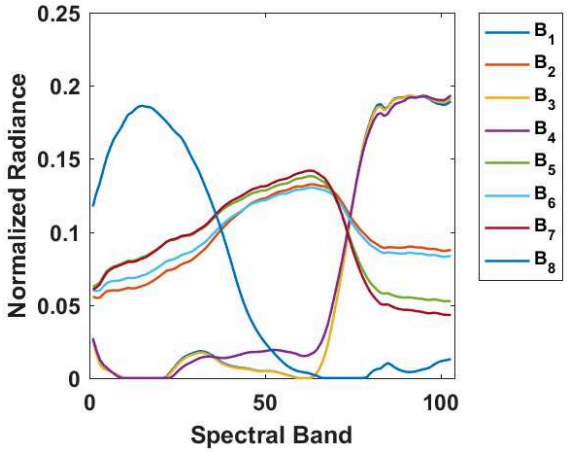
Figure 5.20: Pavia. NCPD results of the MM-HSI tensor for $R = 4$

NMF. Indeed, $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$ look the same, but in fact, one of them is *a slightly shifted version* of the other. Regarding the spectral components, they reflect their corresponding abundance maps with significantly very low SAD values. Considering the similarity between $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$, we notice that $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$ almost coincide. Finally, the most interesting part is perhaps the plot of $\boldsymbol{C}$. Initially, one might expect the curves to be straight since, quantitatively, the collective variability is supposed to be the same for patches. However, while it shows that $\boldsymbol{C}_1$ and $\boldsymbol{C}_4$ are almost straight, it is not the case for $\boldsymbol{C}_2$ and $\boldsymbol{C}_3$ separately, but collectively it is. For example, we see that where one of them is high, the other is low. In part, this observation means that $\boldsymbol{C}_2$ represents the shifts between $k = 1$ and $k = 5$, and $\boldsymbol{C}_3$ represents those between $k = 6$ and $k = 9$. In another part, the two components fluctuate in a way that *reflects the constant variability* (i.e. *to maintain the quantitative balance*) across the frontal slices.

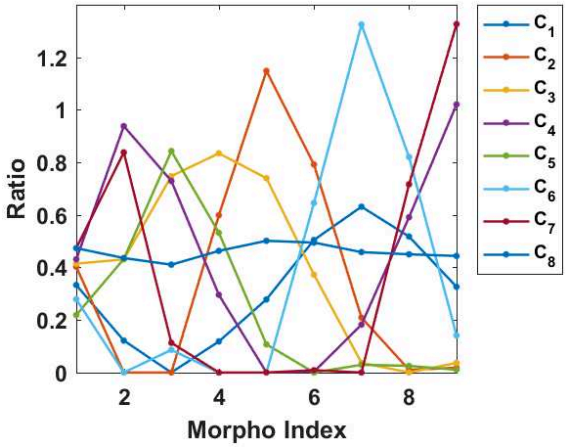
Looking at figure 5.20, we see that $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$ represent *Trees* and *Metal Sheets* and can be compared to those obtained by NMF. As for $\boldsymbol{A}_1$ and $\boldsymbol{A}_4$, they have more interesting features that clearly reflect the "morphological awareness" of the spatial regularization property of RegELMM: $\boldsymbol{A}_1$ highlights additional areas like asphalt roads and bare soil with respect to the



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$



(c) Components of $\boldsymbol{C}$

Figure 5.21: Pavia. NCPD results of the Patch-HSI tensor for $R = 8$

cases of patches and NMF, and $\boldsymbol{A}_4$ clearly shows the shadow areas, which is not the case for patches and NMF. The corresponding spectral and morphological patterns can be explained as was done in section 5.4.3.2. Moreover, the SAD values are significantly very low.

Looking at figure 5.21, the same comments as those of $R = 4$ can be made but with more duplicated maps. The sets of duplicates are $\{\boldsymbol{A}_1, \boldsymbol{A}_3, \boldsymbol{A}_4\}$ and $\{\boldsymbol{A}_2, \boldsymbol{A}_5, \boldsymbol{A}_6, \boldsymbol{A}_7\}$.

5.4.4 Results - Urban

In this section, we present the experiments carried out on the Urban HSI using NMF and NCPD. We go through the same order as the case of Pavia. In general, we notice that the observations are the same as Pavia's, so in order to avoid repetition, we briefly go over the results. The MM-HSI tensor is built using the same technique but with SE of successive radii (in pixels): $\{1, 4, 7, 10\}$, which corresponds to $K = 9$ frontal slices shown in figure 5.22. The patch-HSI tensor is built using 3×3 patches, which corresponds to $K = 9$ frontal slices shown in figure 5.23. Consequently, both data tensors have the dimensions $94249 \times 162 \times 9$. As for the ranks, we also choose the values $R = 4$ and $R = 8$ for NMF and NCPD respectively.

Figure 5.22: False colors of the transformations that form the EMP of the MM-HSI tensor of Urban. The differences across the slices are clearly noticeable. This supports the physical significance of MM for unmixing in RegELMM (i.e., if we project these slices on figure 5.7, we can tell how the variability function, through $\boldsymbol{\Psi}_{(k)}$, will be affected by the morphological characteristics).

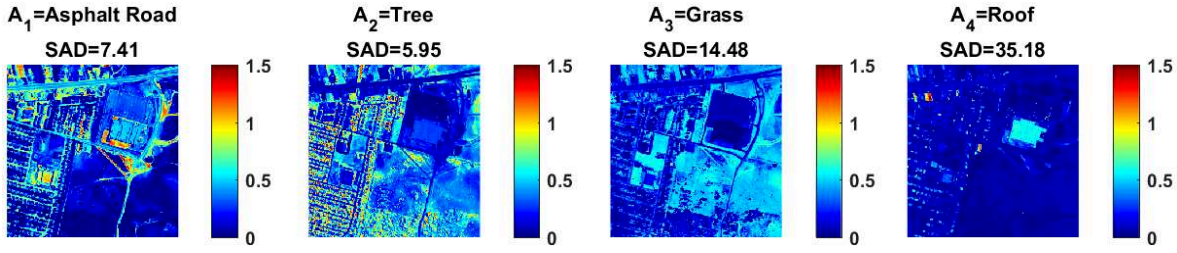
Figure 5.23: False colors of the slices that form the 3×3 patch-HSI tensor of Pavia. The shifts across the slices are not noticeable, so they look visually the same. This is why patches have no physical significance for unmixing in RegELMM (i.e., if we project these slices on figure 5.7, we can tell how the variability function through $\boldsymbol{\Psi}_{(k)}$ represents almost the same information for each slice).

5.4.4.1 NMF Results

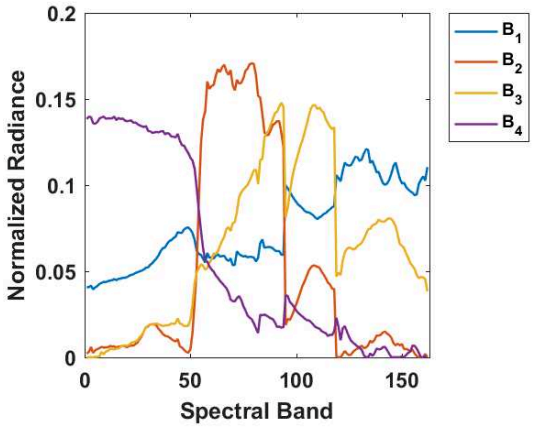
Figure 5.24 shows the abundance maps and spectral signatures of $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively. We obtain four components with relatively low SAD values and good abundance similarity with respect to the reference. We just note that some dark areas like asphalt roads are not highlighted.

5.4.4.2 NCPD Results

Figure 5.25 shows the components of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$. To begin with, we note that the reference spectra of *Tree* and *Grass* are close. In the following, we group the discussion from the perspective of figure 5.25a. Regarding figures 5.25b and 5.25c, the comments on the spectral and morphological patterns are the same as those made for Pavia, but briefly speaking, the SAD



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$

Figure 5.24: Urban. NMF results (with ASC) of the matricized HSI for $R = 4$

values are either lower or comparable to those of NMF, and the plots reflect the qualitative features that appear in their corresponding abundance maps.

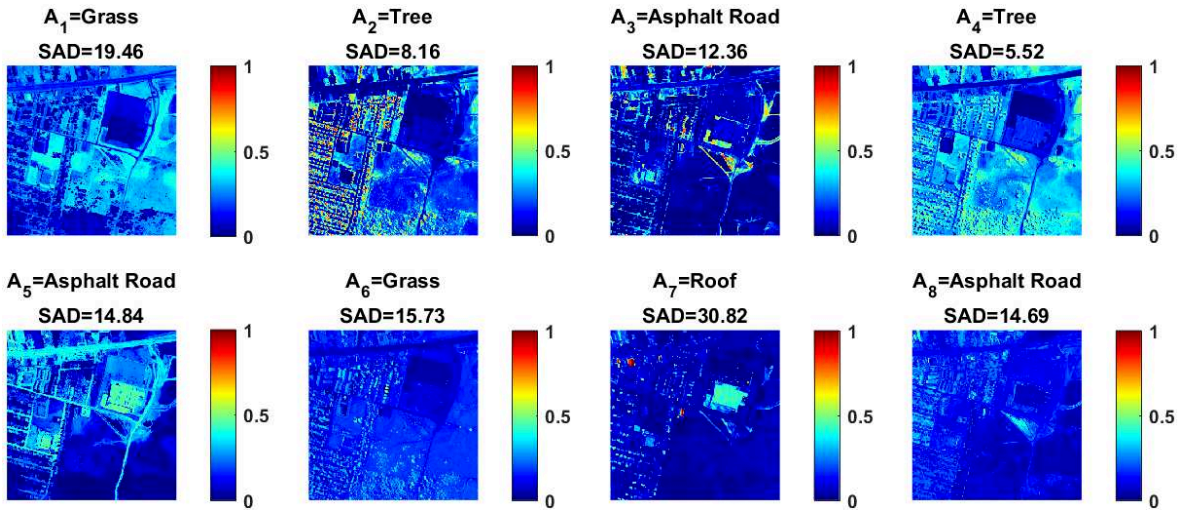
$\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_4$, and $\boldsymbol{A}_6$ were identified as *Tree* and *Grass*. $\boldsymbol{A}_1$ highlights grass areas and fields which looks similar to the *Grass* component of NMF. $\boldsymbol{A}_2$ and $\boldsymbol{A}_4$ respectively highlight small and big tree areas and together they correspond to the *Tree* component of NMF. $\boldsymbol{A}_6$ highlights dark vegetation areas like shadows (we can see that through $\boldsymbol{C}_6$ as well).

$\boldsymbol{A}_3$, $\boldsymbol{A}_5$, and $\boldsymbol{A}_8$ are identified as *Asphalt Road*. $\boldsymbol{A}_8$ seems to correspond to shadow features (judging from $\boldsymbol{C}_8$), but it is not certain, so we skip it. $\boldsymbol{A}_3$ highlights small road areas such as dirt and disconnected narrow streets, while $\boldsymbol{A}_5$ highlights large road areas like the main road and the connected pathways. Here, the asphalt roads are better highlighted, but we notice that the big roof is also highlighted (in $\boldsymbol{A}_5$), which appears in the case of NMF but not in the abundance reference.

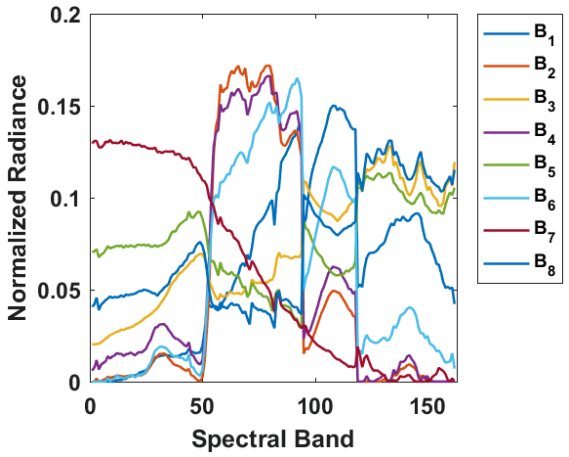
$\boldsymbol{A}_7$ is identified as *Roof* and highlights both small and large building roofs.

5.4.4.3 ELMM Analysis, Comparison Between Patches and MM tensors

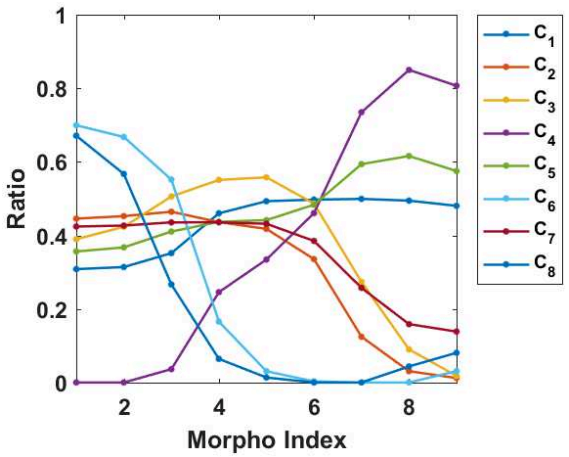
Having the same interests and setup as the case of Pavia, we just note that here we present the results only for $R = 4$ to avoid repetition. That said, figures 5.26 and 5.27 represent



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$



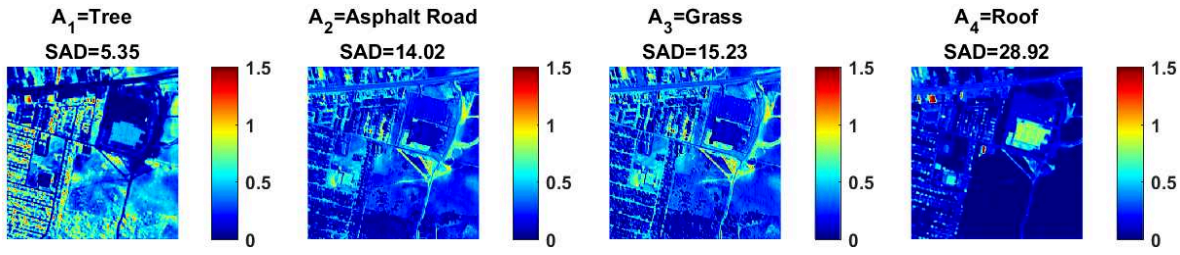
(c) Components of $\boldsymbol{C}$

Figure 5.25: Urban. NCPD results (with ASC) of the MM-HSI tensor for $R = 8$

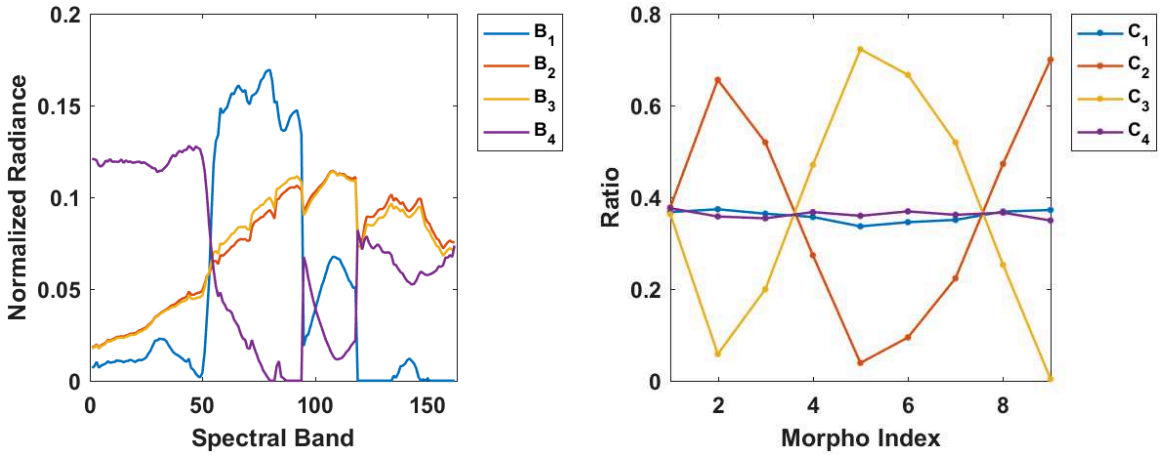
patch-HSI and MM-HSI NCPD respectively. For both figures, we notice a significant drop in the SAD values of *Roof* compared to NMF, and that the big roof feature appears only faintly in another component.

Looking at figure 5.26, first, we notice that the abundance map for *Grass* is missing and that the asphalt road is not highlighted well. $\boldsymbol{A}_1$ and $\boldsymbol{A}_4$ represent *Tree* and *Roof* respectively, and $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$ represent *Asphalt Road* and are "duplicates", which reflects in the signatures of $\boldsymbol{B}_2$ and $\boldsymbol{B}_3$. Moreover, we can see the *constant variability* (i.e. *quantitative balance*) patterns in the plot of $\boldsymbol{C}$ that was observed in the case of Pavia.

On the other hand, looking at figure 5.27, $\boldsymbol{A}_1$ highlights the roads and partially some grass areas, $\boldsymbol{A}_2$ highlights trees and partially some grass areas, and $\boldsymbol{A}_3$ highlights the roofs. As for $\boldsymbol{A}_4$, it highlights shadows of buildings and trees that fall on grass areas, which is also observed in $\boldsymbol{C}_4$.



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$

(c) Components of $\boldsymbol{C}$

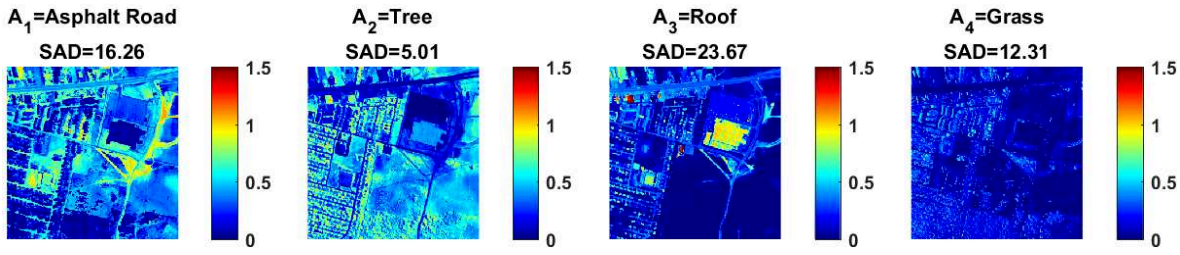
Figure 5.26: Urban. NCPD results of the Patch-HSI tensor for $R = 4$

5.5 Conclusion

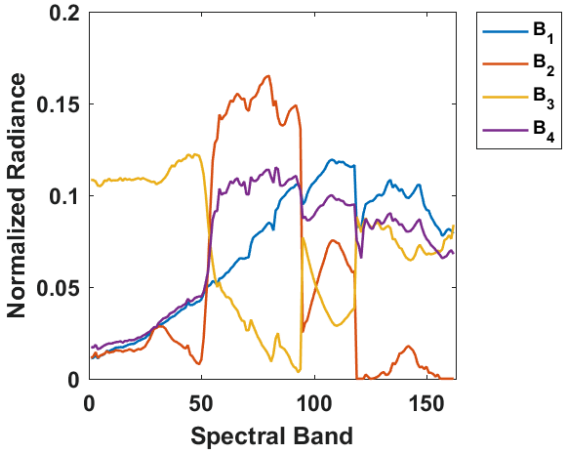
In this chapter, we performed high-order spectral-spatial unmixing through tensor CP decomposition, which has a definition in the framework of ELMM, with MM as an example to extract successive spatial features of the pixels like scale and brightness. It is an application where multiple fields meet.

We explored the quality of unmixed components and compared them to those of NMF and patch-tensor NCPD. In general, unmixing using NMF and patch-NCPD was based only on spectral features, while that of MM-NCPD was based jointly on spectral and morphological ones.

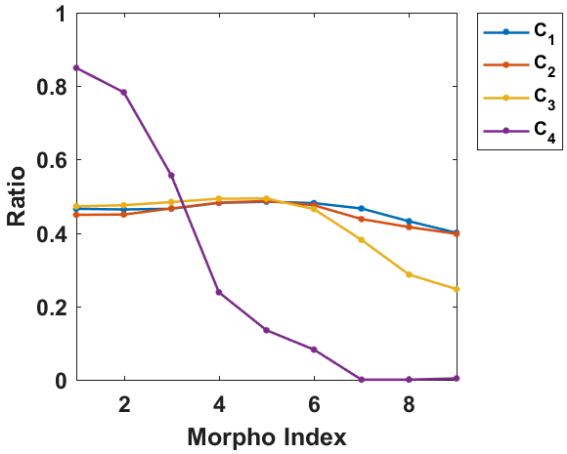
Moreover, we explored the properties of RegELMM where the third-mode diversity of the tensor has physical significance, and compared them to the cases of NMF and patches. In general, MM-NCPD was able to highlight certain parts of the scene better than the other cases, and this aspect was demonstrated for different values of the rank. Briefly speaking, it is possible to benefit from (a) a multi-feature unmixing of the data thanks to tensor analysis, (b) a general framework for CPD defined in ELMM to model spectral variability based on the type of third-mode diversity, and (c) a spatial regularization that is aware of the physical spatial features especially when the imposed rank is low.



(a) Components of $\boldsymbol{A}$



(b) Components of $\boldsymbol{B}$



(c) Components of $\boldsymbol{C}$

Figure 5.27: Urban. NCPD results of the MM-HSI tensor for $R = 4$

However, there are some challenges here. First, compared to NMF, observing the fractional sum-to-one abundances of the reference spectra is not straightforward as the interpretation of the unmixed components is not restricted to spectral features. For that, we note that it can become easier when the columns of $\boldsymbol{B}$ are thought of as double-tagged according to the information provided by their corresponding columns in $\boldsymbol{C}$, so, as a simple example, instead of two *Vegetation* components, we have one *Vegetation/Small* and another *Vegetation/Large* as illustrated in figure 5.8c. Second, having an artificial third-mode diversity such as MM requires setting the parameters of the transformations, which would be easier if one has prior knowledge regarding the physical properties of the scene. For that, we gave a remark by the end of Section 5.2.

Finally, the assumption of the tensor rank being low is another challenge to be dealt with. As mentioned in Chapter 2, the estimation of the rank is a hard problem, but in a current work that is still ongoing, we will discuss the case of a HSI tensor that is built from sequential filtering of the HSI with an example on MM and patches. For instance, the original HSI and its transformations may contain a lot of correlated information along the third mode, almost analogous to how the column vectors of the spectral mode of a single HSI can be very correlated. As such, if the transformations add new intrinsic information, such as the case of MM, the rank of the data may indeed increase compared to that of the original HSI, but only slightly. In a current work, we plan to reason on this assumption through some existing bounds on the tensor rank as well as by looking at the subspaces obtained through SVD.

Chapter 6

Conclusions and Perspectives

Sommaire

6.1 Summary

In this manuscript, we started with a general informal introduction to the different topics and areas that surround the title of this thesis. This introduction hopefully serves as a smooth way to invite interested researchers and individuals into the analysis of images based on tensor representations. Briefly speaking, the goal is to explore the theoretical and applicative involvement of multi-linear algebra in the data mining of multi-modal images. This allows to define a multi-linear framework for high-order image processing for the future as high-order data are becoming ubiquitous lately, and in case where they are not, this framework may encourage the high-order acquisition and representations of such data. Within the scope of this thesis, a particular emphasis was made on (1) Hyperspectral Images (HSIs), among other types of images, for their importance in defining materials through the natural acquisition of their spectral features, and (2) constructing tensors with additional diversities based on artificial extraction of spatial features in the data, specifically through Mathematical Morphology (MM), for their advantages of having low-rank tensor decomposition (under certain circumstances) with unique approximations of the extracted materials.

In general terms, while working on and exploring the different applications of multi-modal image processing (e.g., compression, scene classification, and spectral unmixing), we encountered some challenges related to tensor decomposition and tried to deal with them. In the following, we recall and recount these challenges and give remarks on how they were addressed and treated. After that, we talk about some possible perspectives and open questions.

6.1.1 Challenges and Perspectives

After the general introduction, in Chapter 2, we provided a literature review in an attempt to compile the different tensor decomposition techniques, definitions, uniqueness conditions, algorithms, and their applications in remote sensing images in a proper format hoping that it equips the readers with enough references and resources in order to form a solid idea of the different existing challenges and advance from there. In particular, the selected applications in Section 2.5 put some of the effectiveness of tensor techniques for image analysis into perspective when compared to other approaches.

However, some tensor techniques and their corresponding applications in image analysis were not highlighted, even though they are related to image analysis, since they are not directly related to the contributions of this thesis and in order to keep the manuscript self-contained and focused as much as possible. For instance, Tensor Train (TT) decomposition [76] and Tensor Ring (TR) decomposition [114] (a special case of TT decomposition) were only briefly mentioned in Section 2.4 along with some of their applications in image analysis such as the use of TT decomposition for image recognition [13] and that of TR decomposition for hyperspectral image compressive sensing reconstruction [18]. Moreover, tensor decomposition is used in some works of Satellite Aperture Radar (SAR) image data such as in Multipass SAR Interferometry [62], but since the thesis did not practically deal with SAR applications, the latter was not mentioned. As such, we plan in the future to integrate the aforementioned works and expand on some others, all into a self-contained tutorial contribution in order to highlight the effectiveness of tensor representations for image processing community. After the literature review, we went through the works that were carried out during the thesis, which presented us with a set of challenges that we needed to deal with.

In Chapter 3, we were working with the algorithm ProCo ALS which is an algorithm proposed for the fast computation of CP decomposition of big data tensors under hard-thresholding constraints such as nonnegativity. We noticed that in the case of nonnegative constraints, which is important in image processing, a problem was recurring when completely negative columns were projected into the nonnegative quadrant as zero-columns, which in turn become undefined and discarded after normalizing the columns. For that, we proposed a solution by separating the outer products that form the decomposable tensors into negative and nonnegative counterparts while preserving the result of the outer product itself, and proposed a greedy practical solution for it as the original solution becomes exhaustive when the rank increases. By doing so, we showed that rank-1 tensors can also be seen as individual entities as well as a collection of vectors in an outer product.

Automatic estimation of the tensor rank. Here, we note that finding the exact tensor rank is still a problem in the literature, and in many cases, it boils down to trial and error. As such, sometimes the tensor rank can be *overestimated*. In the application of Chapter 3, and while looking at Figure 3.2, as part of the perspectives and open challenges, one could investigate a trade-off between allowing some "bad" columns to be discarded while observing that the plot of the reconstruction error still decreases healthily, which means that the overestimated rank automatically decreases while we know that there is still enough degrees of freedom to

ideally decompose the data. For instance, in ideal tensor data which do not contain noise, if the plot of the error converges at some high value (compared to zero) like the case of the blue and red plots of the same figure, it means that the current rank (or in other words, the number of columns in the factor matrices) is insufficient to fully represent the data in the CP decomposition.

Presence of noise. In the presence of noise, the estimation of the rank becomes trickier. In such cases, one could use a combination of HOSVD and the potentially proposed method in order to first remove the noisy components and then try to automatically estimate it with a reasonable error plot.

In Chapter 4, we dealt with supervised HSI scene classification using MM, where morphological spatial features are extracted in order to enhance the quality of the classification. Compared to classical approaches where the spectral and spatial feature modes are either treated separately or merged into one, we showed how Nonnegative CP decomposition of tensors built with spectral and morphological features can be used as a direct tool that *jointly* provides a matrix of pixels and features that can be passed to a classifier. In this work, we carried out experiments on real HSIs with tensors of orders 3 and 4 and showed how the spectral-spatial multi-feature relationships can still be intuitively represented, interpreted, and classified as features obtained through a relatively low-rank CP decomposition compared to state-of-the-art methods. At this stage, we adopted AO-ADMM for its efficiency and flexibility with constraints, but the challenge was to add compression for the sake of big image data due to the huge number of pixels in real HSIs, for which we proposed a way to compute CP decomposition using AO-ADMM algorithm with nonnegativity and compression constraints in a similar way to ProCo ALS.

Low-rank assumption. An open question in this work is to reason on the low-rank assumption of the constructed tensors since the value of the rank basically defines the dimension of the feature space, i.e., the number of columns of the factor matrix. In other words, are we sure that a HSI tensor constructed by sequential morphological filtering is low-rank as we claim it to be? If yes, how do we know that? In this work specifically, we avoided getting into the algebraic reasoning and only referred to the tensor rank being low by relatively keeping it less than the dimensions of the feature spaces of the state-of-the-art approaches, which were analytically possible to obtain (through SVD and HOSVD). In scene classification, the rank is eventually perceived as the reduced dimensionality value of the feature space that is finally processed by the classifier, where values such as 30 and 40 can be seen as "low" compared to data with thousands of features (for instance, in the case of Pavia University HSI, we have $103 * 13 = 1339$ total spectral and morphological features). In the following, we give some notes on where this can be a problem especially for solution uniqueness.

Solution Uniqueness. In practice, one could simply pass any input value of the rank to the CP decomposition and still obtain extracted materials that somehow look interpretable. For example, we carried out experiments with different values of the rank varying from 10 to 40 and still obtained generally increasing values of classification accuracy, noting that this increase becomes almost saturated as we go higher. As long as the number of extracted features (the rank) is not over-estimated, the higher it is, the better it becomes for supervised

classification since the data will be trained anyways. However, it is one thing to pass a value and assume that it is low from the application's point of view, and another thing to reason on the assumption itself algebraically then determine a value accordingly. For instance, while the classification accuracy gave better values in general with tensor ranks such as 30 and 40, we might be risking the solution uniqueness since the number of columns in some factor matrices becomes much greater than the number of rows (for Pavia HSI, the third factor matrix has dimensions 13×30 or 13×40). Since the application of scene classification is important, it could add a sense of stability when uniqueness is guaranteed. One direct way to do so is to impose sparsity with nonnegativity as explained in Chapter 2.

A hard way is to approximate the tensor rank algebraically as much as possible, which is a potential open question and one that is essential not only to blind separation or unsupervised classification, but also to other applications where tensors are involved. We reason on this point, in a work that is currently in development, through the bounds of the tensor rank and by comparing the low-dimensional sample and feature subspaces of the different slices that form the tensor, in which a third way is obtained as a transformation of the two-way data. The findings on the rank and properties of the transformations used would be relevant also to other types of data, not only hyperspectral images.

Mastering the choice of parameters. Another practical problem when it comes to scene classification and MM is the choice of the parameters of the classifier itself and the morphological operators. This primarily depends on the structures present in the data, but it is also indirectly related to tensor decomposition since the parameters are remotely related to the quantity of intrinsic information (rank and uniqueness) and the quality or interpretation of the results (extracted materials and properties). It could be a possibility to find a clear connection between the parameters beforehand.

Multi-feature Unmixing (blind unmixing and unsupervised classification). A natural extension of the application of supervised scene classification is that of blind unmixing and unsupervised classification of the data. In this regard, after showing that CP decomposition and MM are good candidates to enhance the classification accuracy of materials in a supervised framework, we would like to see their effectiveness in the case of blind extraction of materials and features, which better showcases the strength of CP decomposition. For that, we worked on Chapter 5.

In Chapter 5, we generalized the problem of multi-feature hyperspectral unmixing by exploring the different aspects of the relationship between CP decomposition and ELMM. There, we emphasized through Mathematical Morphology on the importance of having a third-mode that represents physically meaningful features, which allows to extract materials with spectral and morphological patterns and deals with spectral variability within clusters of pixels that spatially belong to the same objects. Moreover, we faced a challenge to incorporate Abundance Sum-to-one Constraints (ASC) in AO-ADMM for the CP decomposition of third-order tensors, for which we proposed a practical implementation and an interpretation of what it means to have the abundances sum to one in a multi-feature fashion as shown through the Figures of the simplexes of RegELMM. Here, we also talked about how the number of frontal slices K and the rank R influence the result of unmixing. We note that the choice of the

rank in unmixing is more critical than that of supervised classification since we usually do not or barely have any prior information, and thus we care that we extract a small number of components and identify them.

Other types of filters. In the perspectives, one could try other filters as additional diversities or spatial regularization depending on the application (Gaussian filtering, wavelet analysis, etc) and the interpretation of their features, and thus explore their properties in terms of tensor decomposition like the case with time-series, patches, and MM.

6.1.2 Other Potential Works

This area of research has more potential to grow. In the following, we give different suggestions on what areas can be investigated between tensor decomposition and image analysis from combined methodological and applicative points of view.

For instance, with the advancement and huge popularity and domination of machine and deep learning techniques almost everywhere especially recently with image processing and some familiar applications that were mentioned here, one way could be by investigating the involvement or representations of tensor techniques through such methods. There are already some very recent works and resources that deal with this area that can be exploited such as:

- Tensor decomposition for signal processing and machine learning [90].
- Information Geometry, where it would be possible to manipulate manifolds and subspaces extracted from tensor data, which would extends to subspace-based learning and clustering for example [47, 46].
- Tensor Networks, where different tensor decomposition techniques along with their parameters are translated into networks of nodes and networks. Some examples on images for dimensionality reduction can be found in the books [23, 24].
- Some works in Feedforward Neural Network (FNN) such as the classification of HSIs through tensor-based learning models by training the data through the building blocks of the CP decomposition (that is, the vectors of the outer products) [70].
- Tensor libraries intended to grow for machine learning such as TensorLy, which is built in Python [66]. In fact, we prepared a course on feature extraction with matrix and tensor techniques for the doctoral school "Data Science for Geoscience 2020" by using notebooks prepared with the library of TensorLy.

Another way can be through exploring other tensor decomposition techniques such as BTD, PARALIND, CONFAC, Tensor Trains, among others. It is mentioned that BTD is important when it comes to absorbing the high rankness of the spatial structure of original 3-way HSIs, which for instance may work better than CP decomposition in the work of FNN for HSI classification [70] by lowering the number of trained components. The main methodological

and practical challenge of BTD may be to determine the size of the partitioning of the factor matrices beforehand, which could be listed as potential perspectives. Up to our knowledge, there are probably no applications of PARALIND and CONFAC for image analysis, but in general, their models are potentially important when the priority is to extract patterns from the data where it is also desired to completely avoid collinearity of the columns in the factor matrices. Their main challenge is that the set of constrained matrices can be hard to fix or estimate beforehand.

Bibliography

[1] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. Sousa. Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry. *Remote Sensing*, 9(11):1110, 2017.

[2] H. Akbari, K. Uto, Y. Kosugi, K. Kojima, and N. Tanaka. Cancer detection using infrared hyperspectral imaging. *Cancer science*, 102(4):852–857, 2011.

[3] J. Amigo. *Hyperspectral Imaging*. ISSN. Elsevier Science, 2019.

[4] E. Aptoula, M. Dalla Mura, and S. Lefèvre. Vector attribute profiles for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3208–3220, 2016.

[5] L. Armi and S. Fekri-Ershad. Texture image analysis and texture classification methods-A review. *arXiv preprint arXiv:1904.06554*, 2019.

[6] J. Benediktsson, J. Palmason, and J. Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491, 2005.

[7] J. Benediktsson, M. Pesaresi, and K. Amason. Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9):1940–1949, 2003.

[8] J. A. Benediktsson and P. Ghamisi. *Spectral-spatial classification of hyperspectral remote sensing images*. Artech House, 2015.

[9] A. Bhaskara, M. Charikar, and A. Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. In V. F. M. Balcan and C. Szepesvari, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proc. Machine Learning Research*, pages 742–778, Barcelona, Spain, June 2014. PMLR. arXiv:1304:8087.

[10] J. M. Bioucas, A. Plaza, N. Dobligeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE J. Selected Topics Appl. Earth Observ. Remote Sensing*, 5(2):354–379, 2012.

[11] G. Blekherman and Z. Teitler. On maximum, typical and generic ranks. *Mathematische Annalen*, 362(3-4):1021–1031, 2015.

[12] C. Bocci, L. Chiantini, and G. Ottaviani. Refined methods for the identifiability of tensors. *Annali di Matematica Pura ed Applicata (1923-)*, 193(6):1691–1702, 2014.

[13] D. Brandoni and V. Simoncini. Tensor-Train decomposition for image recognition. *Calcolo*, 57(1):9, 2020.

[14] D. Brie, S. Miron, F. Caland, and C. Mustin. An uniqueness condition for the 4-way CANDECOMP/PARAFAC model with collinear loadings in three modes. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4108–4111. IEEE, 2011.

[15] R. Bro, R. A. Harshman, N. D. Sidiropoulos, and M. E. Lundy. Modeling multi-way data with linearly dependent loadings. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 23(7-8):324–340, 2009.

[16] A. M. Bruckstein, M. Elad, and M. Zibulevsky. A non-negative and sparse enough solution of an underdetermined linear system of equations is unique. *IEEE Trans. Inf. Theory*, 54(11):4813–4820, 2008.

[17] G. Camps-Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(6):1351–1362, 2005.

[18] Y. Chen, T.-Z. Huang, W. He, N. Yokoya, and X.-L. Zhao. Hyperspectral image compressive sensing reconstruction using subspace-based nonlocal tensor ring decomposition. *IEEE Transactions on Image Processing*, 2020.

[19] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54(10):6232–6251, 2016.

[20] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected topics in applied earth observations and remote sensing*, 7(6):2094–2107, 2014.

[21] L. Chiantini and G. Ottaviani. On generic identifiability of 3-tensors of small rank. *SIAM Journal on Matrix Analysis and Applications*, 33(3):1018–1037, 2012.

[22] L. Chiantini, G. Ottaviani, and N. Vannieuwenhoven. An algorithm for generic and low-rank specific identifiability of complex tensors. *SIAM J. matrix Ana. Appl.*, 35(4):1265–1287, 2014.

[23] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.

[24] A. Cichocki, A.-H. Phan, Q. Zhao, N. Lee, I. V. Oseledets, M. Sugiyama, and D. Mandic. Tensor networks for dimensionality reduction and large-scale optimizations: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 9(6):431–673, 2017.

[25] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[26] J. Cohen. *Environmental multiway data mining*. PhD thesis, University Grenoble Alpes, 2016.

[27] J. Cohen, R. Farias, and P. Comon. Fast decomposition of large nonnegative tensors. *IEEE Signal Processing Letters*, 22(7):862–866, 2015.

[28] J. Cohen and U. Rothblum. Nonnegative ranks, decompositions and factorizations of nonnegative matrices. *Lin. Alg. Appl.*, 190:149–168, 1993.

[29] P. Comon. Tensors: a brief introduction. *IEEE Sig. Proc. Magazine*, 31(3):44–53, May 2014. hal-00923279.

[30] P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. Academic Press, Oxford UK, Burlington USA, 2010. hal-00460653.

[31] M. Dalla Mura, J. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute profiles for the analysis of very high resolution images. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3747–3762, 2010.

[32] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute filters for the analysis of very high resolution remote sensing images. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 3, pages III–97. IEEE, 2009.

[33] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. Extended profiles with morphological attribute filters for the analysis of hyperspectral data. *International Journal of Remote Sensing*, 31(22):5975–5991, 2010.

[34] A. L. de Almeida, G. Favier, and J. C. M. Mota. A constrained factor decomposition with application to MIMO antenna systems. *IEEE Transactions on Signal Processing*, 56(6):2429–2442, 2008.

[35] L. De Lathauwer. Decompositions of a higher-order tensor in block terms Part II: Definitions and uniqueness. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1033–1066, 2008.

[36] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[37] I. Domanov and L. De Lathauwer. On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part I: Basic results and uniqueness of one factor matrix. *SIAM Journal on Matrix Analysis and Applications*, 34(3):855–875, 2013.

[38] I. Domanov and L. De Lathauwer. On the uniqueness of the canonical polyadic decomposition of third-order tensors—Part II: Uniqueness of the overall decomposition. *SIAM Journal on Matrix Analysis and Applications*, 34(3):876–903, 2013.

[39] L. Drumetz, J. Chanussot, and C. Jutten. Variability of the endmembers in spectral unmixing: recent advances. In *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–5. IEEE, 2016.

[40] L. Drumetz, J. Chanussot, and C. Jutten. Spectral unmixing: A derivation of the extended linear mixing model from the hapke model. *IEEE Geoscience and Remote Sensing Letters*, 2019.

[41] L. Drumetz, M. Dalla Mura, G. Tochon, and R. Fablet. Learning endmember dynamics in multitemporal hyperspectral data using a state-space model formulation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2483–2487. IEEE, 2020.

[42] L. Drumetz, M.-A. Veganzones, S. Henrot, R. Phlypo, J. Chanussot, and C. Jutten. Blind hyperspectral unmixing using an extended linear mixing model to address spectral variability. *IEEE Transactions on Image Processing*, 25(8):3890–3905, 2016.

[43] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2012.

[44] B. Feng and J. Wang. Hyperspectral image dimension reduction using weight modified tensor-patch-based methods. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:3367–3380, 2020.

[45] L. Gao and R. T. Smith. Optical hyperspectral imaging in microscopy and spectroscopy–a review of data acquisition. *Journal of biophotonics*, 8(6):441–456, 2015.

[46] B. B. Gatto, E. M. dos Santos, A. L. Koerich, K. Fukui, and W. S. Junior. Tensor analysis with N-mode generalized difference subspace. *Expert Systems with Applications*, 171:114559, 2021.

[47] B. B. Gatto, M. A. Molinetti, E. M. dos Santos, and K. Fukui. Tensor fukunaga-koontz transform for hierarchical clustering. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 150–155. IEEE, 2019.

[48] Y. Gu, T. Liu, and J. Li. Superpixel tensor model for spatial-spectral classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[49] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Series in Computational Mathematics. Springer, Berlin, Heidelberg, 2012.

[50] L. He, J. Li, C. Liu, and S. Li. Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines. *IEEE Transactions on Geoscience and Remote Sensing*, 56(3):1579–1597, 2017.

[51] C. J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

[52] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al. *A practical guide to support vector classification*, 2003.

[53] K. Huang, N. Sidiropoulos, and A. Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, 2016.

[54] J. R. Jensen. *Remote sensing of the environment: An earth resource perspective 2/e*. Pearson Education India, 2009.

[55] I. Jolliffe. *Principal component analysis*. Springer, 2011.

[56] M. Jouni, M. Dalla Mura, and P. Comon. Classification of hyperspectral images as tensors using nonnegative CP decomposition. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 189–201. Springer, 2019.

[57] M. Jouni, M. Dalla Mura, and P. Comon. Hyperspectral image classification based on mathematical morphology and tensor decomposition. *Mathematical Morphology - Theory and Applications*, 1, 2019.

[58] M. Jouni, M. Dalla Mura, and P. Comon. Hyperspectral image classification using tensor CP decomposition. In *2019 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2019, Yokohama, Japan, July 28 - August 2, 2019*, pages 1164–1167, 2019.

[59] M. Jouni, M. Dalla Mura, and P. Comon. Hyperspectral multi-feature unmixing using tensor decomposition. Manuscript submitted for publication, 2020.

[60] M. Jouni, M. D. Mura, and P. Comon. Some issues in computing the CP decomposition of nonnegative tensors. In *14th Int. Conf. on Latent Variable Analysis and Signal Separation (LVA-ICA)*, Univ. of Surrey, Guildford, UK, July 2-6 2018. Springer. hal-01784370.

[61] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W.-K. Ma. Hyperspectral superresolution: A coupled tensor factorization approach. *IEEE Transactions on Signal Processing*, 66(24):6503–6517, 2018.

[62] J. Kang, Y. Wang, and X. X. Zhu. Multipass SAR interferometry based on total variation regularized robust low rank tensor decomposition. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[63] A. Karami, M. Yazdi, and G. Mercier. Compression of hyperspectral images using discerete wavelet transform and tucker decomposition. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5(2):444–450, 2012.

[64] N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE signal processing magazine*, 19(1):44–57, 2002.

[65] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[66] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic. Tensorly: Tensor learning in python. *The Journal of Machine Learning Research*, 20(1):925–930, 2019.

[67] J. B. Kruskal. Three-way arrays: Rank and uniqueness of trilinear decompositions. *Linear Algebra and Applications*, 18:95–138, 1977.

[68] J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. AMS publ., 2012.

[69] L.-H. Lim and P. Comon. Nonnegative approximations of nonnegative tensors. *Jour. Chemometrics*, 23:432–441, Aug. 2009. hal-00410056.

[70] K. Makantasis, A. Georgogiannis, A. Voulodimos, I. Georgoulas, A. Doulamis, and N. Doulamis. Rank-R FNN: A tensor-based learning model for high-order data classification. Manuscript submitted for publication, 2020.

[71] P. R. Marpu, M. Pedergnana, M. Dalla Mura, S. Peeters, J. A. Benediktsson, and L. Bruzzone. Classification of hyperspectral data using extended attribute profiles based on supervised and unsupervised feature extraction techniques. *International Journal of Image and Data Fusion*, 3(3):269–298, 2012.

[72] S. Murchie, R. Arvidson, P. Bedini, K. Beisser, J.-P. Bibring, J. Bishop, J. Boldt, P. Cavender, T. Choo, R. Clancy, et al. Compact reconnaissance imaging spectrometer for mars (CRISM) on mars reconnaissance orbiter (MRO). *Journal of Geophysical Research: Planets*, 112(E5), 2007.

[73] L. Najman and H. Talbot. *Mathematical morphology: from theory to applications*. John Wiley & Sons, 2013.

[74] T. Nouri, M. M. Oskouei, B. Alizadeh, P. Gamba, and A. Marinoni. Improvement of the MVC-NMF problem using particle swarm optimization for mineralogical unmixing of noisy hyperspectral data. *Journal of the Indian Society of Remote Sensing*, 47(4):541–550, 2019.

[75] S. Ono and T. Kasai. Efficient constrained tensor factorization by alternating optimization with primal-dual splitting. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3379–3383. IEEE, 2018.

[76] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[77] C. Pilorget and J.-P. Bibring. NIR reflectance hyperspectral microscopy for planetary science: Application to the MicrOmega instrument. *Planetary and Space Science*, 76:42–52, 2013.

[78] R. Plemmons and R. Cline. The generalized inverse of a nonnegative matrix. *Proceedings of the american mathematical society*, pages 46–50, 1972.

[79] C. Prévost, K. Usevich, P. Comon, and D. Brie. Hyperspectral super-resolution with coupled tucker approximation: Recoverability and SVD-based algorithms. *IEEE Transactions on Signal Processing*, 2020.

[80] Y. Qi, P. Comon, and L.-H. Lim. Semialgebraic geometry of nonnegative tensor rank. *SIAM J. Matrix Ana. Appl.*, 37(4):1556–1580, Nov. 2016. arxiv:1601.05351.

[81] Y. Qi, P. Comon, and L. H. Lim. Uniqueness of nonnegative tensor approximations. *IEEE Trans. Inf. Theory*, 62(4):2170–2183, Apr. 2016. arXiv:1410.8129.

[82] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly. Hyperspectral unmixing via $l_{1/2}$ sparsity-constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4282–4297, 2011.

[83] C. Rodarmel and J. Shan. Principal component analysis for hyperspectral image classification. *Surveying and Land Information Science*, 62(2):115–122, 2002.

[84] J. R. Ruiz-Tolosa and E. Castillo. *From Vectors to Tensors*. Universitext. Springer, Berlin, Heidelberg, 2005.

[85] J. Serra. *Image Analysis and Mathematical Morphology*. Number v. 1 in Image Analysis and Mathematical Morphology. Academic Press, 1984.

[86] J. Serra. *Image analysis and mathematical morphology: Theoretical advances*. Image Analysis and Mathematical Morphology. Academic Press, 1988.

[87] G. A. Shaw and H. K. Burke. Spectral imaging for remote sensing. *Lincoln laboratory journal*, 14(1):3–28, 2003.

[88] C. Shi and L. Wang. Incorporating spatial information in spectral unmixing: A review. *Remote Sensing of Environment*, 149:70–87, 2014.

[89] N. D. Sidiropoulos and R. Bro. On the uniqueness of multilinear decomposition of N-way arrays. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):229–239, 2000.

[90] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

[91] P. Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[92] M. Sørensen and L. De Lathauwer. Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms—part I: Uniqueness. *SIAM Journal on Matrix Analysis and Applications*, 36(2):496–522, 2015.

[93] A. Stegeman and A. L. De Almeida. Uniqueness conditions for constrained three-way factor decompositions with linearly dependent loadings. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1469–1490, 2010.

[94] A. Stegeman and N. D. Sidiropoulos. On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition. *Linear Algebra and its applications*, 420(2-3):540–552, 2007.

[95] V. Strassen. Rank and optimal computation of generic tensors. *Linear algebra and its applications*, 52:645–685, 1983.

[96] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson. SVM-and MRF-based method for accurate classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 7(4):736–740, 2010.

[97] D. Tuia, F. Pacifici, M. Kanevski, and W. J. Emery. Classification of very high spatial resolution imagery using mathematical morphology and support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 47(11):3866–3879, 2009.

[98] K. Uto. Estimation of crop vitality based on tensor decomposition and data fusion of multimodal, multitemporal leaf-scale aerial images. `https://www.jst.go.jp/kisoken/presto/en/project/1112075/1112075_14.html`.

[99] K. Uto, M. Dalla Mura, Y. Sasaki, and K. Shinoda. Estimation of leaf angle distribution based on statistical properties of leaf shading distribution. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 5195–5198. IEEE.

[100] K. Uto, M. D. Mura, and J. Chanussot. Normal direction and true color estimation of leaves based on tensor decomposition of leaf-scale optical images. In *Whispers 2018 - Poster Session*, 2018.

[101] K. Uto, M. D. Mura, and J. Chanussot. Spatial resolution enhancement of optical images based on tensor decomposition. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, jul 2018.

[102] M. A. Veganzones, J. E. Cohen, R. C. Farias, J. Chanussot, and P. Comon. Nonnegative tensor CP decomposition of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 54(5):2577–2588, 2016.

[103] M. A. Veganzones, J. E. Cohen, R. C. Farias, K. Usevich, L. Drumetz, J. Chanussot, and P. Comon. Canonical polyadic decomposition of hyperspectral patch tensors. In *2016 24th European Signal Processing Conference (EUSIPCO)*, pages 2176–2180. IEEE, 2016.

[104] M. A. Veganzones, L. Drumetz, G. Tochon, M. Dalla Mura, A. Plaza, J. Bioucas-Dias, and J. Chanussot. A new extended linear mixing model to address spectral variability. In *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2014.

[105] S. Velasco-Forero and J. Angulo. Classification of hyperspectral images by tensor modeling and additive morphological decomposition. *Pattern Recognition*, 46(2):566–577, 2013.

[106] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE transactions on image processing*, 2(2):176–201, 1993.

[107] L. Wang, C. Shi, C. Diao, W. Ji, and D. Yin. A survey of methods incorporating spatial information in image classification and spectral unmixing. *International Journal of Remote Sensing*, 37(16):3870–3910, 2016.

[108] F. Xiong, Y. Qian, J. Zhou, and Y. Tang. Hyperspectral unmixing via total variation regularized nonnegative tensor factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 2018.

[109] Y. Xu, Z. Wu, J. Chanussot, P. Comon, and Z. Wei. Nonlocal coupled tensor CP decomposition for hyperspectral and multispectral image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[110] Z. Yang, Y. Zhang, W. Yan, Y. Xiang, and S. Xie. A fast non-smooth nonnegative matrix factorization for learning sparse representation. *IEEE access*, 4:5161–5168, 2016.

[111] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang. Blind spectral unmixing based on sparse nonnegative matrix factorization. *IEEE Transactions on Image Processing*, 20(4):1112–1125, 2010.

[112] N. Yokoya, C. Grohnfeldt, and J. Chanussot. Hyperspectral and multispectral data fusion: A comparative review of the recent literature. *IEEE Geoscience and Remote Sensing Magazine*, 5(2):29–56, 2017.

[113] N. Yokoya, T. Yairi, and A. Iwasaki. Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 50(2):528–537, 2011.

[114] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

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

Résumé — Nous considérons une image dans laquelle chaque pixel n est défini par un vecteur $\boldsymbol{y}_{(n)}$ de dimension m, contenant m observations d'une quantité variable. Cette variable est mesurée séquentiellement et pourrait être le spectre de la lumière (par exemple, les images hyperspectrales), le temps (c'est-à-dire une vidéo), différents angles d'acquisition, etc. Par exemple, une image RVB est composée de trois (c'est-à-dire $m = 3$) des canaux adjacents dans le domaine spectral s'étendant approximativement à des longueurs d'onde rouges à bleues. Il est souvent significatif d'exprimer ce vecteur comme une combinaison linéaire de la forme compacte comme: $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}$ où $\boldsymbol{Y}$, $\boldsymbol{X}$ et $\boldsymbol{A}$ sont de dimension $m \times n$, $m \times p$ et $p \times n$. Il est clair qu'avec cette écriture, la position exacte des pixels n'est pas prise en compte, ni l'ordre des valeurs mesurées. Même la taille de l'image n'est pas explicite; Si l'image est $n_1 \times n_2$, seul le produit $n = n_1 n_2$ apparaît effectivement. Plus important encore, si les pixels et les variables mesurées sont permutés, les rangées de $\boldsymbol{X}$ et les colonnes de $\boldsymbol{A}$ sont permutées en conséquence. L'un des objectifs de ce sujet de recherche est de corriger ces indéterminations, car les permutations sont pertinentes (c'est-à-dire que la position des pixels et l'ordre séquentiel des valeurs sont des caractéristiques significatives). Au moins deux tentatives peuvent être trouvées dans la littérature. Cette idée a commencé à être étudiée dans le cadre d'un stage au cours de l'été 2017 à GIPSA-Lab.

Mots clés : Décomposition de tenseurs, Factorisation matricielle, Images hyperspectrales, Télédétection, Morphologie mathématique, Données multimodales, Algèbre tensorielle, Algèbre multilinéaire.

Abstract — We consider an image in which every pixel n is defined by a vector $\boldsymbol{y}_{(n)}$ of dimension m, containing m observations of a varying quantity. This variable is measured sequentially and could be the spectrum of light (e.g., hyperspectral images), time (i.e., a video), different angles of acquisition etc. For example, a RGB image is composed of three (i.e., $m = 3$) channels adjacent in the spectral domain ranging from approximately red to blue wavelengths. It is often meaningful to express this vector as a linear combination of the compact form as: $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{A}$ where $\boldsymbol{Y}$, $\boldsymbol{X}$ and $\boldsymbol{A}$ are of dimension $m \times n$, $m \times p$ and $p \times n$. It is clear that with this writing, the exact position of pixels is not taken into account, nor is the order of the measured values. Even the size of the image is not explicit; if the image is $n_1 \times n_2$, only the product $n = n_1 n_2$ indeed appears. More importantly, if pixels and measured variables are permuted, rows of $\boldsymbol{X}$ and columns of $\boldsymbol{A}$ are permuted accordingly. One of the goals of this research topic is to fix these indeterminacies, because permutations are relevant (i.e., the position of pixels and sequential order of values are meaningful features). At least two attempts can be found in the literature. This idea has started to be investigated in the framework of an internship during Summer 2017 at Gipsa-Lab.

Keywords: Tensor Decomposition, Matrix Factorization, Hyperspectral Images, Remote Sensing, Mathematical Morphology, Multi-modal Data, Tensor Algebra, Multi-linear Algebra.

Grenoble Images Parole Signal Automatique Laboratoire (GIPSA-Lab), 11 Rue des Mathématiques, 38400 Saint-Martin-d'Hères, France
Saint Martin d'Hères