THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à MINES ParisTech

Spatial machine learning applied to multivariate and multimodal images

Ecole doctorale n°432

SCIENCES DES METIERS DE L'INGENIEUR

Spécialité MORPHOLOGIE MATHEMATIQUE

COMPOSITION DU JURY :

Mme. Isabelle BLOCH
Institut Mines-Telecom, Telecom ParisTech, Présidente du jury

M. Gustau CAMPS-VALLS
Universitat de València, Rapporteur

M. Ludovic MACAIRE
Université Lille1, Rapporteur

M. Stéphane MALLAT
Ecole normale supérieure, PSL Research University, Membre du jury

M. Maxime MOREAUD
IFP Energies nouvelles, Membre du jury

M. Jesús ANGULO
Mines ParisTech, Membre du jury

M. Loïc SORBIER
IFP Energies nouvelles, invité

Soutenue par Gianni FRANCHI
le 21 septembre 2016

Dirigée par **Jesús Angulo**

MINES ParisTech | PSL RESEARCH UNIVERSITY PARIS

Remerciements

J'aimerais tout d'abord remercier mon directeur de thèse, Jesus Angulo, pour son encadrement hors pair. Ainsi que pour la confiance qu'il m'a accordée en acceptant de m'encadrer, et pour toutes les heures qu'il a consacrées à me diriger, à m'aider et à me conseiller durant mes recherches. Sa grande disponibilité et ainsi que nos nombreux échanges m'ont permis de mieux définir mon projet. Je souhaite également remercier Maxime Monreaud et Loïc Sorbier, mes encadrants à IFP Energie Nouvelles, pour leur aide et leurs conseils tout au long de ma thèse, ainsi que leur implication tout au long de mon projet qui a permis de rendre possible beaucoup de nos idées.

Ce travail n'aurait pas été possible sans les Mines ParisTech, IFP Energie Nouvelles ainsi que de l'association des anciens des Mines ParisTech, qui grâce à leur soutien financier m'ont permis de me consacrer sereinement à mes travaux de recherches.

Je remercie également les membres de mon jury de thèse pour avoir formulé des critiques constructives et pour leurs conseils. Je remercie Isabelle Bloch d'avoir accepté de présider ce dernier, ainsi que Gustau Camps-Valls et Ludovic Macaire pour avoir accepté d'être les rapporteurs de mon manuscrit. Ma reconnaissance se porte aussi à l'attention de Stéphane Mallat qui a accepté d'être membre de mon jury de thèse.

Je remercie de plus l'ensemble des personnes qui ont contribué d'une manière ou d'une autre à ce projet de thèse. Je n'aurais jamais pu réaliser ce travail doctoral sans les formations fournies par les Mines ParisTech ainsi que celles du Master MVA de l'ENS Cachan. Je souhaite donc remercier l'ensemble des professeurs ou membres du personnel de ces formations avec qui j'ai intéragi.

Pendant mon doctorat j'ai eu l'occasion de faire un échange au département de statistique de l'université d'Oxford. Je souhaiterais remercier mes collègues de l'université d'Oxford, et en particulier Dino Sejdinovic pour avoir accepté de m'encadrer, pour sa confiance ainsi que pour toute l'aide qu'il m'a fourni.

De plus, durant ces trois ans sur le site des Mines de Paris à Fontainebleau j'ai eu l'occasion de cotoyer de nombreux collègues. J'aimerais remercier en particulier mes collègues de Géostatistics et CBIO, et en particulier Emilie Chautru, Christian Lantuejoul ainsi que Jean-Philippe Vert pour avoir pris du temps pour répondre à toutes mes questions.

Je remercie ensuite les membres du CMM. Je tiens à les remercier pour leur accueil, leur aide, ainsi que pour les differents échanges que j'ai eu avec eux qui ont été une source d'enrichissement, à la fois sur le plan scientifique et humain. Je tiens à remercier: Michel Bilodeau, Serge Beucher, Etienne Decencière, Petr Dokladal, Dominique Jeulin, Beatriz Marcotegui, Fernand Meyer, François Willot, Bruno Figliuzzi, Santiago Velasco-Forero, Serge Koudoro, Matthieu Faessel. Un grand merci également à Catherine Moysan et à Anne-Marie de Castro pour leur accompagnement dans toutes les démarches administratives et leur bienveillance.

Je remercie enfin les thésards et post-doctorants que j'ai cotoyé au Mines ParisTech Fontainebleau ainsi qu'à Paris. Je les remercie pour leur générosité, leur bonne humeur, leur amitié, leur cours de salsa, leurs tricheries répétées en soirée jeu, leur manque de talent culinaire (surtout quand il s'agit de faire des pâtes), leurs soirées crêpes excellentes, leurs soirées repas censées commencer à 20:30 mais qui finalement

commencent à 23:30, leurs critiques intempestives sur ma façon de conduire, leur mauvais niveau en ski, de m'avoir enseigné l'escalade, et enfin pour m'avoir oublié à IKEA.... Pour tout ça je remercie Amin, Aurélie, Bassam, Benjamin, Daniele, Elise, Emmanuel, Enguerrand, Haisheng, Joris, Jean Baptiste, Jean-Charles, Lydia, Luc, Mauro, Sébastien, Serge, Simona, Théodore et Vaïa. Vous avez su rendre magique mon expérience à Fontainebleau.

Je remercie également tous mes autres amis : Anais, Basile, Bastien, Franck, Giancarlo, Guillaume, Jeremy, Laura, Laurent, Marco, Marion, Xavier.

Enfin, je remercie l'ensemble des personnes de ma famille qui m'ont soutenu au cours de mes études et dans mon projet de thèse.

Abstract

This thesis focuses on multivariate spatial statistics and machine learning applied to hyperspectral and multimodal and images in remote sensing and scanning electron microscopy (SEM). In this thesis the following topics are considered:

Fusion of images:
SEM allows us to acquire images from a given sample using different modalities. The purpose of these studies is to analyze the interest of fusion of information to improve the multimodal SEM images acquisition. We have modeled and implemented various techniques of image fusion of information, based in particular on spatial regression theory. They have been assessed on various datasets.

Spatial classification of multivariate image pixels:
We have proposed a novel approach for pixel classification in multi/hyperspectral images. The aim of this technique is to represent and efficiently describe the spatial/spectral features of multivariate images. These multi-scale deep descriptors aim at representing the content of the image while considering invariances related to the texture and to its geometric transformations.

Spatial dimensionality reduction:
We have developed a technique to extract a feature space using morphological principal component analysis. Indeed, in order to take into account the spatial and structural information we used mathematical morphology operators

Keywords

Image processing, Machine learning, Kernel methods, Mathematical morphology, Principal Component Analysis, Support Vector Machine, Deep learning, Scattering transform, Kriging

Résumé

Cette thèse porte sur la statistique spatiale multivariée et l'apprentissage appliqués aux images hyperspectrales et multimodales. Les thèmes suivants sont abordés :

Fusion d'images :
Le microscope électronique à balayage (MEB) permet d'acquérir des images à partir d'un échantillon donné en utilisant différentes modalités. Le but de ces études est d'analyser l'intérêt de la fusion de l'information pour améliorer les images acquises par MEB. Nous avons mis en oeuvre différentes techniques de fusion de l'information des images, basées en particulier sur la théorie de la régression spatiale. Ces solutions ont été testées sur quelques jeux de données réelles et simulées.

Classification spatiale des pixels d'images multivariées :
Nous avons proposé une nouvelle approche pour la classification de pixels d'images multi/hyper-spectrales. Le but de cette technique est de représenter et de décrire de façon efficace les caractéristiques spatiales / spectrales de ces images. Ces descripteurs multi-échelle profond visent à représenter le contenu de l'image tout en tenant compte des invariances liées à la texture et à ses transformations géométriques.

Réduction spatiale de dimensionnalité :
Nous proposons une technique pour extraire l'espace des fonctions en utilisant l'analyse en composante morphologiques. Ainsi, pour ajouter de l'information spatiale et structurelle, nous avons utilisé les opérateurs de morphologie mathématique.

Mots Clés

Traitement de l'image, Machine Learning, Méthodes à noyaux, Morphologie mathématique, Analyse en Composantes Principales, Support Vector Machine, Apprentissage profond, Transformée de scattering, Krigeage.

Table of Contents

Part I

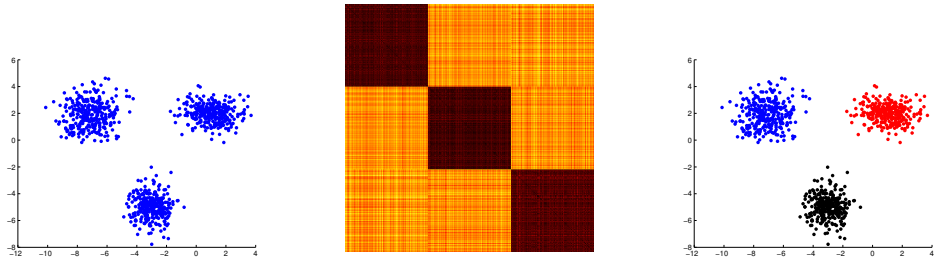
Introduction

Multivariate images

A grey scale image can be seen as a matrix in which each pixel is an entry of the matrix. Such an image brings information on the spatial structures present in it; a natural evolution was to add the color information. On color images each pixel is a vector of size 3. Thanks to these images we have both spatial and spectral information. However the spectral information is quite limited since we just have access to the colors visible by human eye. This can be sufficient in some cases, but if we want to be more precise we might need more spectral information. That consideration gave birth to hyperspectral images. On each pixel of these images we have a vector which is a signal that represents the reflected light of the area under study. Color or hyperspectral images are examples of multivariate images. Multivariate images are images where each pixel is described by a vector of size superior to 1, contrary to traditional univariate (or scalar) images. These types of images are becoming more important because of the sensors abilities and can arise from a huge variety of sources. Multivariate images appear in remote sensing, astronomy, molecular imaging, etc. The image analysis of these sets must be different, and adapted to the nature of the data itself. In this thesis we will work with two kind of multivariate images. On the one hand, on multispectral and hyperspectral images in Part 1. Typical examples of these images are obtained by remote sensing devices, and our objective is mainly to classify each pixel of them, using the spectral and spatial information. The second kind of data we are interested in is the Energy Dispersive Spectrometry (EDX), that are used in physicochemical characterization of materials. The work on these images are the fruit of a collaboration between MINES ParisTech and IFP Energies Nouvelles.

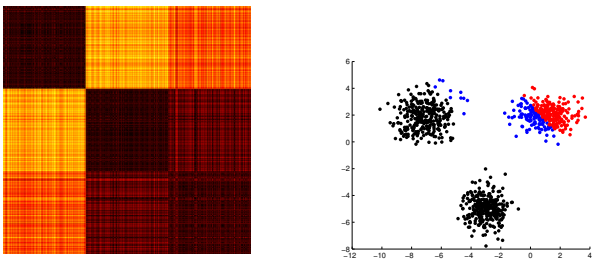
Curse of dimensionality

The curse of dimensionality is a term invented by Richard Bellman [13] which designates the phenomena that occur when data are studied in a space of high dimension. Suppose that the data can be represented by a vector $v \in [0,1]^D$, where D is the dimension of the data space. Consider first a simple case where $D = 1$, in this case it is very easy to calculate that with 100 points, you can get an interval between the points which is about 10^{-2}. However, if we consider a space of dimension $D = 10$ and if you want the items to be separated by a ball of radius 10^{-2}, we easily see that it takes 10^{20} points. Now, if we consider a space of dimension D, we see that we need 10^{2D} points which can be astronomical. This shows that to fight the curse of dimensionality we must increase the number of data. However most of the time the number of data is limited. Let us consider that we have 900 vectors v_i $i \in [1, 900]$ of dimension 2, see example in Figure 1(a). These vectors are the result of a Gaussian mixture model, where three Gaussians have been considered. To study these vectors, we plot in Figure 1(b) the distance matrix d, which is a square matrix of size 900 such that $d_{i,j} = \|v_i - v_j\|_2$. Then, using the k-means clustering algorithm [48], we can naturally separate the data into three clusters, which is illustrated in Figure 1(c). When the dimension of the data is equal to 2, it seems easy to analyze the data. However if the dimension of the data is increased, as in Figure 2(a)(b), we can see that the distance matrix d is not relevant any more for $D = 100$ and so the k-means is not able to find the three clusters.



(a) (b) (c)

Figure 1: A set of 900 data points of dimension $D = 2$. In (a) the data, in (b) their $d_{i,j}$ distance matrix, in (c) the $k = 3$ clusters of the dataset obtained by k-means.



(a) (b)

Figure 2: A set of 900 data points of dimension $D = 100$. In (a) the $d_{i,j}$ distance matrix of the dataset. As we can see, it is more difficult to separate some classes. In (b), the $k = 3$ clusters of the data obtained by k-means: clustering is not correct because of the curse of dimensionality.

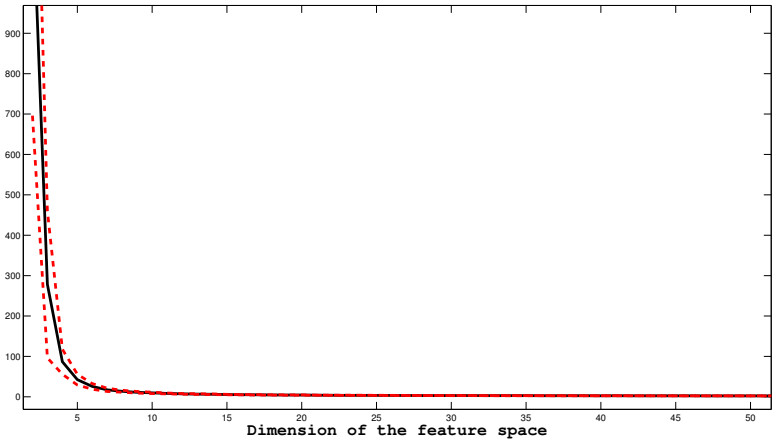


Figure 3: In this figure we have selected randomly 500 data points, $v_i \in [0,1]^D$, $i = 1, \cdots, 500$, where D is the dimension of the feature space. We plot $R = \frac{\max_{(i,j)}\{\|v_i - v_j\|_2\}}{\min_{(i,j)}\{\|v_i - v_j\|_2\}}$, which represents the power of discrimination of the distance. To have consistent results, 100 Monte Carlo simulations have been generated, and the black curve represents the mean result, and the dash-dotted lines represent the mean $+/-$ the standard deviation.

Moreover, it can be shown (see [55]) that because of the dimension D of the space, the ratio between the maximum Euclidean distance and the minimum Euclidean distance, i.e.,

$$R = \frac{\max_{(i,j)}\{\|v_i - v_j\|_2\}}{\min_{(i,j)} \|v_i - v_j\|_2\}},$$

of the data tends to 1 when D increases. Hence, the euclidean distance has a very poor discriminating power with high dimensional data. We represented this lost of discriminant power of the L_2 norm in Figure 3.

Indeed, in the case of ultraspectral images, D is usually about one or two thousand, which shows how sparse the sampled manifold is (or how empty the whole space is), and how crucial is the question about similarity between spectra. Therefore, due to the high dimensionality of the space, it is critical to learn a good representation in order to fight against the curse of dimensionality.

There is the range of approaches belonging to the dimensionality reduction paradigm, so one could work on a smaller dimension space and thus circumvents the curse of dimensionality. It is also possible to work on techniques able to learn the feature space such as kernel learning methods, or the Convolutional Neural Network (CNN) deep learning approaches.

Multimodal images

Sometimes the acquisition of samples can be done using simultaneously different techniques, which leads to the notion of multimodal image acquisition. These different modalities might bring different information sources that could be combined. This is the case for instance in satellite remote sensing (panchromatic images +

ultra/hyper-spectral images), in biomedical microscopy (multispectral images + Raman spectroscopy image + quantitative phase) or in scanning electron microscopy (secondary electrons/backscattered electrons + EDX). The goal of multimodal information combination can be approached by different information fusion techniques. This kind of data brings new issues; firstly, they must be perfectly registered or in other words, these various images should be spatially consistent: pixel locations from one modality should be related to the other one. In this thesis, we consider that the data are perfectly registered. Secondly the information fusion should be compatible in terms of their ranges. Finally, some modalities might see objects absent in others.

Context of the study on multimodal SEM

IFP Energies Nouvelles is a public-sector research institute, active in the fields of energy, transport and environment. Among the many areas of expertise, IFP Energies nouvelles contributes to mastering and developing microscopic analysis techniques. Energy Dispersive Spectroscopy (EDX) images are more and more important in microanalysis. With this type of picture, it is possible to see in each pixel the composition of the pixel, and thus theoretically to have the composition of the analyzed sample. This type of image could therefore greatly improve the microscopic analysis of a sample by adding new information. It should be noted, that the resolution of these are often poorer than other images from Scanning Electron Microscopy (SEM), and do not take into account the sample 3D structure. So in order to be able to help IFP experimenters to characterize these images we worked, in collaboration with an IFP Energies Nouvelles team, on techniques able to merge the different modalities of the SEM images. The literature on multimodal SEM fusion is, to the best of our knowledge, very limited. Hence, we took inspiration for our techniques from hyperspectral remote sensing images pansharpening techniques.

Spatial representation

We also worked on the classification problem of remote sensing images, which is an important issue on remote sensing.

Multivariate imaging allows us to obtain typically a vector describing each pixel of the image, and thus with this quantity of information acquired, we can in theory determine more easily the class of each pixel. However, as we have discussed above, a major problem in classification is the curse of dimensionality. It is therefore necessary to use techniques of statistical learning in order to extract useful information. At the beginning, linear machine learning technique were the first kinds of techniques used in remote sensing [26, 120]. Unfortunately the nonlinear nature of the data is then completely neglected, so many approaches based on manifold learning methods [149, 127, 12, 39, 29, 132] were developed. However, all these techniques often neglect the interactions between a pixel and its neighbours, which is an essential concept in image processing. Also, these techniques have often been invented for data that are not images and were created to address the problems of "big data" that are nonlinear. Therefore gradually a large number of works have attempted to use spatial descriptors giving birth to spatial machine learning, more naturally adapted to image classification. Among the descriptors based on a combination of

machine learning and image processing, we note the use of mathematical morphology techniques on hyperspectral images to extract interesting features [115, 45, 35]. That brings now one of the best classification results on remote sensing classification. This justifies our choice to work on spatial machine learning.

Main contributions and Thesis outline

The research work presented in this thesis investigates two main topics. The first subject consists in learning appropriate representations for multivariate images. We worked on both dimensionality reduction and classification techniques. The second theme is more domain applied and deals with building efficient representations for the fusion of multimodal SEM images.

This thesis is composed of a series of chapters, most of them are based on the publications done during the three years, and organised as follows:

Part 1: On this part we focus on spatial dimensionality reduction and supervised classification with examples from remote sensing images. First, in Chapter 2 we present techniques able to reduce the dimensionality of the spectra while considering the spatial information. These techniques lean on morphological techniques and are able to learn a manifold that brings spatial and spectral information. In Chapter 3, in order to improve classification of hyperspectral pixels, we propose a deep texture descriptor model able to characterize the hyperspectral in homogenous local areas of the image. This descriptor is based on new innovative techniques, namely the scattering transform [18, 96] and the kernel mean map [140].

Part 2: The second part focusses on multimodal SEM image fusion. Chapter 4 provides a background on the EDX multimodal SEM images. Then, in Chapter 5, we present first some state-of-the-art techniques of pansharpening, and second, after applying these techniques and based on the results, we introduce two techniques based on the bilateral filter and on morphological down/up-sampling theory. Finally, we propose in Chapter 6 a more theoretical study of image fusion based on kriging, also linked with Gaussian process regression.

Part II

Feature representation and classification for hyperspectral images

Chapter 1

State-of-the-Art on Statistical Learning Applied to Hyperspectral Images

Abstract

In machine learning, high-dimensional data correspond to points lying in spaces of dimension typically higher than 10 and are difficult to represent and so to interpret. Due to the curse of dimensionality, feature extraction and classification in such spaces is challenging. A typical approach consists in "simplifying the data", usually by projecting the data onto a space of low dimension. This kind of techniques is commonly followed in hyperspectral image processing. These images come from hyperspectral sensors and can be seen as a three-dimensional hyperspectral data cube where each image represents a spectral band. In general the number of bands is around ten for multispectral images, about one hundred for hyperspectral images, and of thousands for ultraspectral images. So it is necessary to use statistical learning techniques to have a better understanding of this data. In this chapter we will focus on a review of the statistical learning techniques that have been developed, on the one hand, to reduce the dimension of our data, and on the other hand, to classify pixels from these images. Many different dimensionality reduction techniques have been proposed, we focus here on the Principal Component Analysis (PCA), Multi-Dimensional Scaling (MDS), Isomap, Kernel-PCA, Diffusion Maps, Locally linear Embedding (LLE), Laplacian Eigenmaps and Hessian LLE. We also remind some dictionary learning algorithms. Finally we explain linear regression, linear classification, Support Vector Machine (SVM), and Convolutional Neural Network (CNN) for classification.

Résumé

L'étude des données en grande dimension (supérieur à 10) est difficile en raison du fléau de la dimensionnalité. Ainsi l'extraction d'information utile est dure. Une approche typique consiste à simplifier les données en projetant les données sur un espace de plus petite dimension, on parle alors de réduction de la dimension. Ce type de techniques est très utile dans le traitement d'images hyperspectrales. Ces images proviennent de capteurs hyperspectrales et peuvent être considérées comme un cube

de données tridimensionnelles, où chaque image représente une bande spectrale. En général, le nombre de bandes est environ dix pour des images multispectrales, une centaine pour des images hyperspectrales. Il est donc nécessaire d'utiliser des techniques d'apprentissage statistique pour mieux comprendre ces données. Dans ce chapitre, nous nous concentrerons sur un examen des techniques d'apprentissage statistique qui ont été élaborées, afin d'une part, de réduire la dimension de nos données et, d'autre part, de classifier. De nombreuses techniques de réduction de la dimension ont été proposées, nous nous concentrons ici sur l'analyse des composantes principales (APC), multidimensionnelle scalling (MDS), Isomap, Kernel-PCA, diffusion map, Locally Linear Embedding (LLE), Laplacian Eigenmaps and Hessian LLE. Nous rappelons également certains algorithmes d'apprentissage de dictionnaire. Enfin, nous expliquons la régression linéaire, la classification linéare, le Support Vector Machine (SVM) et les Convolutional Neural Networks (CNN) pour la classification.

1.1 Introduction

Conventionally, in remote sensing, and in other image processing domains, we use instruments whose measurements are in optical, infrared and radar electromagnetic spectrum areas. Hyperspectral imaging is an evolution of optical imaging to reconstruct the spectral profiles of objects imaged by the acquisition of several tens or several hundred narrow spectral bands, that totally covers the whole of the optical spectral range, from purple to infrared (in most of the case but we can choose another part of the spectrum). This type of imaging, because of its complexity and its data size, is still limited to mainly experimental applications. However, progress in technology and information begin to turn it more operational. That is why several space agencies develop hyperspectral sensors. Among many other initiatives, we can highlight the current German and Italian programs (EnMAP and PRISMA), the American missions in progress Hyperion and the new program HyspIRI (coming soon), the SPECTRA program for ESA. But these kinds of images provide rich information that there can also be used in medical imaging, in physicochemical imaging, etc. Let us formalize the models for hyperspectral image data representation. We review dimensionality reduction and classification techniques relevant for hyperspectral image analysis.

1.2 Hyperspectral images

In 1666, Isaac Newton showed that light can be decomposed into a spectrum of light rays that can be represented in a graph linking the intensity and the different wavelengths present in a light beam. The objects that we see have colors because they absorb certain wavelengths and reflect others. This reflected light represents what is seen by our eyes. For example: a red apple reflects mainly red light and grass mainly reflects green light.

In the context of imaging, it is the spectrum of reflected light that interests us. However, in traditional imaging we take the wave included in the visible range. Visible light is the part of the electromagnetic radiation that our eyes capture. In fact, the human eye perceives the waves electromagnetic whose wavelengths lie between about 380 and 780 nanometres. However, in the case where the incident

light comes from the sun, by limiting ourselves to the visible spectrum we do not have access to all available information.

Each spectrum has a different name and a different function, such as the spectral range of X-rays used in medical radiology. Thus, depending on its composition, an object absorbs such and such wavelengths and returns a spectrum. This spectrum of reflected light is unique and depends on the composition of the object, and is called spectral signature. Knowledge of the spectral signature of a body can help us to know if it is present or not in a region. It is therefore possible to have access to the material composition.

A hyperspectral image is therefore a simultaneous acquisition of images of many narrow, contiguous spectral bands. This technique enables us to obtain a spectrum for each pixel of the image.

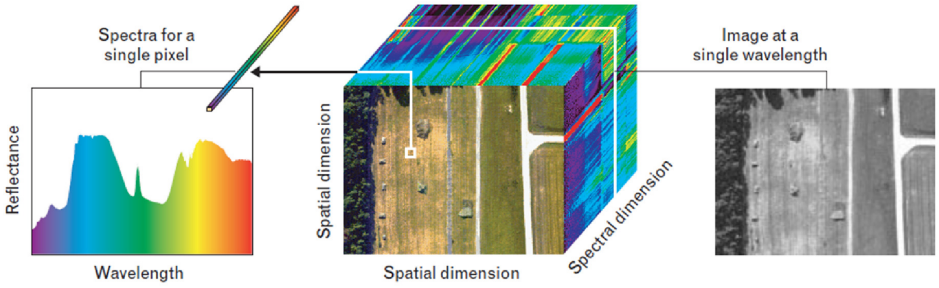


Figure 1.1: Hyperspectral image [98]

1.2.1 Representation of hyperspectral data

Physical description

In this section we describe several physical issues that can deteriorate the acquisition of the data, most of them are mainly related to remote sensing hyperspectral imaging, but some issues are more general.

Atmospheric effects During its path between the object and the satellite, each reflected ray passes through the atmosphere, which changes the reflected spectrum. Indeed, the atmosphere is anisotropic, so there may be interactions between particles within it, through absorption and diffusion phenomena. Therefore, the spectrum recovered from the satellite is not the spectrum of the real object. This is why it is important to consider this aspect, which is one of the main obstacles to hyperspectral satellite imagery. It can be seen by comparing the spectra of Figure 1.2 that the spectrum is actually modified.

Topographic effects Topographic effects are due to the shape of objects. For example in Figure 1.3, we see that the rays A, B and C come from different reliefs and are not reflected in the same place. Thus, it is easy to understand that when a sensor receives a signal from a non-planar surface, a part of the signal received by the sensor does not come from the analyzed area.

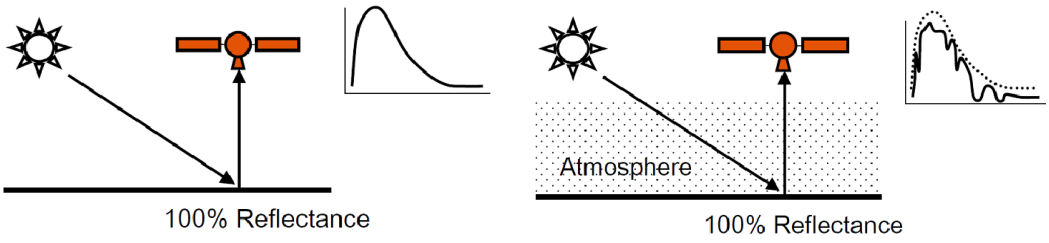


Figure 1.2: Reflected spectra in the left with no atmosphere, in the right in presence of atmosphere [43].

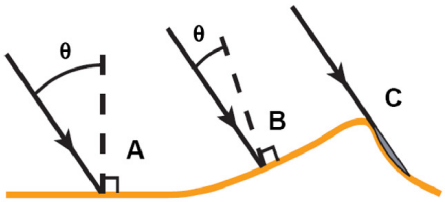


Figure 1.3: Reflected spectra from a non-planar object [43].

Spectral Mixture effects When a photo is taken, the picture is divided into pixels. So that each one has a different color. Even if a pixel is composed of different elements, it will have only one color. However, in hyperspectral images, when a pixel (which is a vector) is composed of different materials we have a spectral mixture. In fact, the spectrum of each pixel is a function of the spectra of different materials integrated in the surface of the detector. This type of problem can make impossible to determine the composition of a pixel.

There are two main models of spectral mixture,

Linear Mixing: in this case we consider that the materials of a pixel are optically separated, thus the spectrum of the pixel is just the sum of the different spectra multiplied by a coefficient related to the proportion materials. In the following we refer to this coefficient as the abundance ratio or abundances.

Nonlinear Mixing : in this case we are dealing with a mixture of different materials that creates dispersion effects. The signal thus obtained is a non linear combination of the different spectral reflections. There are different non linear mixture models according to the non linear aspect we want to represent.

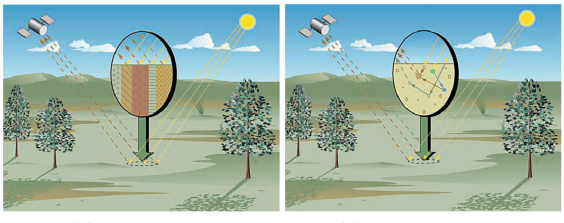


Figure 1.4: The two models of spectral mixing [40].

1.2.2 Mathematical description of linear mixing model

Despite the above description, which shows that in a realistic situation, the received signal by the hyperspectral sensors is nonlinear, we will consider a linear model, as it turns out to be frequently used. This assumption may seems very simplistic, but it is one of the most used models and gives consistent results. Finally, this model also has the advantage of being simple from a mathematical point of view.

The linear mixing model assumes that the D-dimensional spectrum pixel vector v is a linear combination of the R pure spectra, called endmembers, tainted by an additive noise $\mathcal{N}$.

$$v = \sum_{k=1}^{R} \alpha_{x,k}\, \phi_k + \mathcal{N}, \tag{1.1}$$

where $\phi_k \in \mathbb{R}^D$ for $k \in [1, R]$ represent the R endmembers, and $\alpha_{x,k}$ represents the abundance ratio of the endmember ϕ_k over $v = f(x)$ which are constrained:

- all the abundance ratios are positive, i.e., $\alpha_{x,k} \geq 0$, $\forall k$, and
- they belong to the simplex, i.e., $\sum_{k=1}^{R} \alpha_{x,k} = 1$ $\forall x \in E$.

1.3 State-of-the-art on dimensionality reduction

1.3.1 Notation

We introduce here the notation used in the rest of the chapter. Let E be a subset of the discrete space $\mathbb{Z}^2$, which represents the support space of a 2D image and $F \subseteq \mathbb{R}^D$ be a set of pixels values in dimension D. Hence, it is assumed in our case that the value of a pixel $x \in E$ is represented by a vector $v \in F$ of dimension D. This vector v represents the spectrum at the position x. We write $f : E \to F$ for the function associated to the image, so we have $f(x) = v$.

Let F be a set of n spectra such that $F = \{v_i\}_{i=1}^{n} \in \mathbb{R}^D$ and let us assume that these points lie on a smooth manifold $\mathcal{F}$ of intrinsic dimension d with $d \ll D$.

We can consider that F is an hyperspectral image, by doing that, we did not take into account the position of the pixel.

Additionally, we denote the manifolds with a calligraphic upper-case letter ($\mathcal{I}, \mathcal{S}, \ldots$), and $\mathcal{D}$ refers to the dictionary in dictionary learning techniques.

1.3.2 From PCA to manifold learning

The goal of manifold learning techniques is to find a smooth mapping $\Psi : \mathcal{F}' \longrightarrow \mathcal{F}$ where $\mathcal{F}' \subset \mathbb{R}^d$ represents some parameter domain. Mapping Ψ must be found while preserving some characteristic properties. Manifold learning techniques can be mainly divided into two families: i) linear and ii) nonlinear methods. Moreover, the nonlinear techniques can also be divided into global methods that uses only global information from the set of points, and the approaches using local information. We present some of the most popular manifold learning techniques. In particular, we provide more details for PCA and Kernel-PCA, together with MDS.

Principal Component Analysis (PCA) - A linear method

Principal Component Analysis (PCA) has several names, so it is also known as the Karhunen-Loève transform, the Hotelling transform, the Singular Value Decomposition (SVD) transform, etc. Here we will call it PCA.

We start with a set of n points $F = \{v_i\}_{i=1}^n \in \mathbb{R}^D$. The PCA goal is to reduce the dimension of this vector space finding the basis that captures most of the variance of data set thanks to a projection on the principal component space, namely

$$F = \{v_i\}_{i=1}^n \longrightarrow F' = \{v_i'\}_{i=1}^n \tag{1.2}$$

with $v_i' \in \mathbb{R}^d$, where $d \ll D$. A more geometrical point of view of the PCA is to consider that the PCA goal is to find a space of projection such that the mean squared distance between the original vectors and their projection is as small as possible, this is equivalent to finding the projection that maximizes the variance.

Let us call $w_j \in \mathbb{R}^D$ the j principal component. The aim of PCA is to find the set of vectors $\{w_j, 1 \leq j \leq D\}$ such as:

$$\arg\min_{w_j} \left[n^{-1} \sum_{i=1}^{n} \| v_i - < v_i, w_j > w_j \|^2 \right], \quad \forall 1 \leq j \leq D. \tag{1.3}$$

Developing now the distance we have:

$$\| v_i - < v_i, w_j > w_j \|^2 = \| v_i \|^2 - 2 < v_i, w_j >^2 + < v_i, w_j >^2 \| w_j \|^2,$$

then adding the additional constraint that $\|w_j\|^2 = 1$, replacing in (1.3) and keeping only terms that depend on w_j, we have the following new objective function:

$$\arg\max_{w_j, \|w_j\|^2=1} n^{-1} \sum_{i=1}^{n} < v_i, w_j >^2, \quad \forall 1 \leq j \leq D. \tag{1.4}$$

Since

$$\text{var}(< v_i, w_j >) = n^{-1} \sum_{i=1}^{n} (< v_i, w_j >)^2 - (n^{-1} \sum_{i}^{n} (< v_i, w_j >))^2,$$

if we consider that the data F has been column-centered, which means that $\sum_{i=1}^n v_i = 0$, then

$$\text{var}(< v_i, w_j >) = n^{-1} \sum_{i=1}^{n} (< v_i, w_j >)^2.$$

Thus we can see that the goal of the PCA is to find principal components that maximize the variance. The problem can be rewritten in a matrix way using:

$$\begin{aligned} n^{-1} \sum_{i=1}^{n} < v_i, w_j >^2 &= n^{-1} (F w_j)^T (F w_j) \\ &= w_j^T (n^{-1} (F^T F)) w_j = w_j^T V w_j, \end{aligned}$$

where $V = n^{-1}(F^T F)$, $V \in M_{D,D}(\mathbb{R})$, is the covariance matrix of F. Hence we should optimize:

$$\arg\max_{w_j, \|w_j\|^2=1} w_j^T V w_j, \quad \forall 1 \leq j \leq D. \tag{1.5}$$

Thanks to Lagrange multiplier theorem we can rewrite the objective function as:

$$\mathbf{L}(w_j, \lambda) = w_j^T V w_j - \lambda(w_j^T w_j - 1), \tag{1.6}$$

where $\lambda \in \mathbb{R}$. Since we want to maximize this function, we have to derive it and equal it to zero:

$$\frac{\partial \mathbf{L}}{\partial w_j}(w_j, \lambda) = 2Vw_j - 2\lambda w_j = 0.$$

So, we finally obtain as solution

$$Vw_j = \lambda w_j. \tag{1.7}$$

Thus, the principal component w_j that satisfies the objective function is an eigenvector of the covariance matrix V, and the one maximizing $\mathbf{L}(w_j, \lambda)$ is the one with the larger eigenvalue. Then we can have all the w_j by computing the SVD of V.

There are different approaches to choose the reduced dimension d, that is the principal component to be kept. The underlying assumption is the following: if the intrinsic dimension of the data is d, then the remaining $d - D$ eigenvalues, corresponding to the eigenvectors that are discarded, should be significatively small. This principle is expressed using

$$\text{Prop} = \sum_{j=1}^{d} \lambda_j / \sum_{j=1}^{D} \lambda_j,$$

which is equal to the proportion of the original variance kept. We will write W_d the square matrix of size D containing the d eigenvectors corresponding of the higher eigenvalues, and all the other columns are null. Then thanks to the Eckart-Young theorem [165] (theorem 5.1) it is possible to quantify the error of reduction of dimension such as :

$$Err_{\text{PCA}} = \|V - W_d^T V W_d\|_F^2 = \sum_{j=d+1}^{D} \lambda_j^2 \tag{1.8}$$

Multi Dimensional Scaling (MDS) - A linear method

Goal Multidimensional scaling [129] is a data mining technique used to reduce the dimensionality of the data by keeping as close as possible the pairwise distance between the data point. More technically, we have:

$$\mathcal{F} = \{v_i\}_{i=1}^n \longrightarrow \mathcal{F}' = \{v_i'\}_{i=1}^n \tag{1.9}$$

with $\|v_i - v_j\| \simeq \|v_i' - v_j'\|$, $\forall i, j \in [1, n]^2$, where $\|v_i - v_j\|$ represents the Euclidean distance between v_i and v_j. So the goal is to find a configuration of the dataset in a lower dimension such as the similarities are preserved. There are two main objective functions to use in the corresponding optimization problem. The simplest one is the raw stress function, given by:

$$\Phi(\mathcal{F}') = \sum_{i,j \in [1,n]^2} (\|v_i' - v_j'\|_2^2 - \|v_i - v_j\|_2^2), \tag{1.10}$$

The second objective function is the Sammon cost function :

$$\Phi(\mathcal{F}') = \frac{1}{\sum_{i,j\in[1,n]^2} \|v_i - v_j\|_2} \sum_{i,j\in[1,n]^2} (\|v'_i - v'_j\|_2^2 - \|v_i - v_j\|_2^2). \tag{1.11}$$

The unique difference between these two cost functions comes from the term: $\left(\sum_{i,j\in[1,n]^2} \|v_i - v_j\|_2\right.$ that is a way to highlight data points on $\mathcal{F}$ that are more similar.

Mathematical rationale In the following we still represent our initial data by : $\mathcal{F} = \{v_i\}_{i=1}^n \in \mathbb{R}^D$, and we would like to reduce the dimensionality and to build the representation $\mathcal{F}' = \{v'_i\}_{i=1}^n \in \mathbb{R}^d$. First, let us introduce some basic notions.

Definition 1 *The Euclidean distance between two vectors* $a = [a_1, \ldots, a_D] \in \mathbb{R}^D$ *and* $b = [b_1, \ldots, b_D] \in \mathbb{R}^D$ *is just:*

$$d_2(a, b) = \|a - b\|_2 = \sqrt{\sum_{i=1}^{D} (a_i - b_i)^2}.$$

To simplify the writting of this section, we will omit the subscript 2 for the L_2 norm and write just $\|\cdot\|$ instead of $\|\cdot\|_2$. For the rest of this part we will denote the (symmetric) Euclidean distance matrix and the squared Euclidean distance matrix, respectively with matrix D and S such as:

$$D_{i,j} = d_2(v_i, v_j), \qquad S_{i,j} = d_2(v_i, v_j)^2.$$

Definition 2 *The inner product between two vectors* $a = [a_1, \ldots, a_D] \in \mathbb{R}^D$ *and* $b = [b_1, \ldots, b_D] \in \mathbb{R}^D$ *is defined as:*

$$< a, b >= \sum_{i=1}^{D} a_i \times b_i.$$

Since the basic idea of MDS is to transform the distance matrix into a cross product matrix, we are going to introduce the Gram matrix, also called cross product matrix, written G, such as:

$$G_{i,j} =< v_i, v_j > .$$

Another definition of the Gram matrix is the following one.

Definition 3 *Let* X *be a set of data, the Gram matrix* G *associated to* X *is defined by :*

$$G = X^T X$$

Since $d_2(v_i, v_j) = \sqrt{< v_i - v_j, v_i - v_j >}$, we have
$d_2(v_i, v_j) = \sqrt{< v_i, v_i > + < v_j, v_j > -2 < v_i, v_j >}$ thus

$$D_{i,j} = \sqrt{G_{i,i} + G_{j,j} - 2G_{i,j}}.$$

By performing a double centering, we can transform the squared distance matrix into a Gram matrix G. To do it, we need a "centralizing matrix" H such as:

$$H = I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T, \tag{1.12}$$

with $\mathbf{1} \in M_{n,1}(\mathbb{R})$ and $\mathbf{1}^T = [1, \ldots, 1]$.

Lemma 4 *The centralizing matrix H satisfies the following properties:*

(1) $H^2 = H$;

(2) $\mathbf{1}^T H = H\,\mathbf{1} = 0$;

(3) If X is a centred dataset, which means that columnwise mean$(X) = 0$, then $X\,H = X$.

Proof. (1)

$$\begin{aligned} H\,H &= (I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T)\,(I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T), \\ H\,H &= I - \frac{2}{n} \cdot \mathbf{1}\,\mathbf{1}^T + \frac{1}{n^2} \cdot (\mathbf{1}\,\mathbf{1}^T) \times (\mathbf{1}\,\mathbf{1}^T), \\ &\text{however } \mathbf{1}\,\mathbf{1}^T\,\mathbf{1}\,\mathbf{1}^T = n \cdot \mathbf{1}\,\mathbf{1}^T, \\ &\text{so } H\,H = I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T, \\ H\,H &= H. \end{aligned}$$

(2)

$$\begin{aligned} \mathbf{1}^T H &= \mathbf{1}^T\,(I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T), \\ \mathbf{1}^T H &= \mathbf{1}^T - \frac{1}{n} \cdot \mathbf{1}^T\,(\mathbf{1}\,\mathbf{1}^T), \\ &\text{however } \mathbf{1}^T\,\mathbf{1}\,\mathbf{1}^T = n \cdot \mathbf{1}^T, \text{ so } \mathbf{1}^T H = 0, \\ &\text{since } H^T = H, \text{ finally } H\,\mathbf{1} = 0. \end{aligned}$$

(3) If X is a centred dataset, we have $X\,\mathbf{1}\,\mathbf{1}^T = 0$. Then we obtain

$$X\,H = X - \frac{1}{n} \cdot X\,(\mathbf{1}\,\mathbf{1}^T).$$

■

Lemma 5 *Let X be a centred dataset, and G its Gram matrix, then we have*

$$H\,G\,H = G.$$

We add a superscript c on G and X to note that they are centred: G^c and X^c.

Proof. Thanks to the Lemma 4, one has $X^c\,H = X^c$, and using Definition 3, we can write

$$G^c = (X^c)^T\,X^c = H\,(X^c)^T\,X^c \times H = H\,G^c\,H.$$

■

Theorem 6 *Let X be a dataset, and D be its Euclidean distance matrix. We have*

$$D_{i,j} = \sqrt{G^c_{i,i} + G^c_{j,j} - 2G^c_{i,j}}. \tag{1.13}$$

Proof. Let us consider $X = \{x_i\}_{i=1}^n \in \mathbb{R}^k$, $X^c = \{x_i^c\}_{i=1}^n \in \mathbb{R}^k$, D, G^c, receptively a set of data, the same set of data but centred, the Euclidean distance matrix of the set X, the centred Gram matrix of X, and $k \in \mathbb{N}$

$$D_{i,j} = d_2(x_i, x_j) = d_2(x_i - \text{mean}(X), x_j - \text{mean}(X)) = d_2(x_i^c, x_j^c),$$
$$\text{then, one has: } D_{i,j} = \sqrt{< x_i^c - x_j^c, x_i^c - x_j^c >},$$
$$D_{i,j} = \sqrt{< x_i^c, x_i^c > -2 \cdot < x_i^c, x_j^c > + < x_j^c, x_j^c >},$$
$$D_{i,j} = \sqrt{G_{i,i}^c + G_{j,j}^c - 2G_{i,j}^c}.$$

■

Theorem 7 *Let X be a dataset, G its Gram matrix, S its squared distance matrix, centred matrices G^c and S^c obey the following relationship:*

$$G^c = -\frac{1}{2} S^c. \tag{1.14}$$

Proof. Using Lemma 5, we have $H\, G^c\, H = G^c$. Let us translate it matricially, using $G^c = \{G_{ij}^c\}_{i,j \in [1,n]^2}$, to obtain:

$$H\, G^c\, H = \left(I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T\right) G^c \left(I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T\right),$$
$$\text{thus } H\, G^c\, H = G^c - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T\, G^c - \frac{1}{n} \cdot G^c\, \mathbf{1}\,\mathbf{1}^T + \frac{1}{n^2} \cdot \mathbf{1}\,\mathbf{1}^T\, G^c\, \mathbf{1}\,\mathbf{1}^T = G^c,$$
$$\text{hence, } \mathbf{1}\,\mathbf{1}^T\, G^c + G^c\, \mathbf{1}\,\mathbf{1}^T - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T\, G^c\, \mathbf{1}\,\mathbf{1}^T = 0.$$

On the other hand, we have:

$$\mathbf{1}\,\mathbf{1}^T\, G^c = \begin{pmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{pmatrix} \times \begin{pmatrix} G_{11}^c & \dots & G_{1n}^c \\ \vdots & \ddots & \vdots \\ G_{n1}^c & \dots & G_{nn}^c \end{pmatrix},$$

Then by doing the product, we get:

$$\mathbf{1}\,\mathbf{1}^T\, G^c = \begin{pmatrix} \sum_{i=1}^n G_{i1}^c & \dots & \sum_{i=1}^n G_{in}^c \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^n G_{i1}^c & \dots & \sum_{i=1}^n G_{in}^c \end{pmatrix}.$$

In the same way, we can write:

$$G^c\, \mathbf{1}\,\mathbf{1}^T = \begin{pmatrix} \sum_{j=1}^n G_{1j}^c & \dots & \sum_{j=1}^n G_{1j}^c \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^n G_{nj}^c & \dots & \sum_{j=1}^n G_{nj}^c \end{pmatrix}.$$

And also:

$$\mathbf{1}\,\mathbf{1}^T\, G^c \times \mathbf{1}\,\mathbf{1}^T = \begin{pmatrix} \sum_{i,j \in [1,n]^2} G_{ij}^c & \dots & \sum_{i,j \in [1,n]^2} G_{ij}^c \\ \vdots & \ddots & \vdots \\ \sum_{i,j \in [1,n]^2} G_{ij}^c & \dots & \sum_{i,j \in [1,n]^2} G_{ij}^c \end{pmatrix}.$$

So finally we have:

$$\begin{pmatrix} \sum_{i=1}^{n} G^c_{i1} & \dots & \sum_{i=1}^{n} G^c_{in} \\ \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} G^c_{i1} & \dots & \sum_{i=1}^{n} G^c_{in} \end{pmatrix} + \begin{pmatrix} \sum_{j=1}^{n} G^c_{1j} & \dots & \sum_{j=1}^{n} G^c_{1j} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{n} G^c_{nj} & \dots & \sum_{j=1}^{n} G^c_{nj} \end{pmatrix}$$

$$-\frac{1}{n} \cdot \begin{pmatrix} \sum_{i,j\in[1,n]^2} G^c_{ij} & \dots & \sum_{i,j\in[1,n]^2} G^c_{ij} \\ \vdots & \ddots & \vdots \\ \sum_{i,j\in[1,n]^2} G^c_{ij} & \dots & \sum_{i,j\in[1,n]^2} G^c_{ij} \end{pmatrix} = 0_n(\mathbb{R}).$$

By multiplying this expression on the right by $\mathbf{1}$ we obtain:

$$\begin{pmatrix} \sum_{i,j\in[1,n]^2} G^c_{ij} \\ \vdots \\ \sum_{i,j\in[1,n]^2} G^c_{ij} \end{pmatrix} + n \cdot \begin{pmatrix} \sum_{j=1}^{n} G^c_{1j} \\ \vdots \\ \sum_{j=1}^{n} G^c_{nj} \end{pmatrix} - \begin{pmatrix} \sum_{i,j\in[1,n]^2} G^c_{ij} \\ \vdots \\ \sum_{i,j\in[1,n]^2} G^c_{ij} \end{pmatrix} = 0_{n,1}(\mathbb{R})$$

Hence $\forall i \in [1,n]$ we have $\sum_{j=1}^{n} G^c_{ij} = 0$ and thanks to the symmetry of G^c, $\forall j \in [1,n]$ we have $\sum_{i=1}^{n} G^c_{ij} = 0$. In addition, it is known now that $D^2_{i,j} = G^c_{i,i} + G^c_{j,j} - 2G^c_{i,j}$ thanks to Theorem 6, so finally:

$$\sum_{i=1}^{n} D^2_{i,j} = \sum_{i=1}^{n} G^c_{i,i} + n \cdot G^c_{j,j}, \tag{1.15}$$

$$\sum_{j=1}^{n} D^2_{i,j} = n \cdot G^c_{i,i} + \sum_{j=1}^{n} G^c_{j,j}. \tag{1.16}$$

But we also have that

$$S^c = H\,S\,H = \left(I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T\right) S \left(I - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T\right),$$

$$S^c = S - \frac{1}{n} \cdot \mathbf{1}\,\mathbf{1}^T S - \frac{1}{n} \cdot S\,\mathbf{1}\,\mathbf{1}^T - \frac{1}{n^2} \cdot \mathbf{1}\,\mathbf{1}^T S\,\mathbf{1}\,\mathbf{1}^T.$$

By rewriting matricially, and using $S = \{D^2_{ij}\}_{i,j\in[1,n]^2}$ and all the matrix operations that we have done just above, we have:

$$S^c_{ij} = D^2_{ij} - \frac{1}{n} \cdot \sum_{i=1}^{n} D^2_{i,j} - \frac{1}{n} \cdot \sum_{j=1}^{n} D^2_{i,j} + \frac{1}{n^2} \cdot \sum_{i,j\in[1,n]^2} D^2_{ij}. \tag{1.17}$$

Using (1.15) and putting those equations on (1.17), we obtain the following new expression:

$$S^c_{ij} = D^2_{ij} - \frac{1}{n} \cdot \left(\sum_{i=1}^{n} G^c_{i,i} + n \cdot G^c_{j,j} - n \cdot G^c_{i,i} + \sum_{j=1}^{n} G^c_{j,j} - \frac{1}{n} \cdot \sum_{i,j\in[1,n]^2} D^2_{ij} \right)$$

$$S^c_{ij} = D^2_{ij} - G^c_{ii} - G^c_{jj}$$

According to Theorem 6, we have as expected:

$$S^c_{ij} = -2 \cdot G^c_{ij}.$$

■

Definition 8 *The Frobenius norm of a matrix* $A = \{a_{ij}\}_{i,j\in[1,n]^2} \in M_n(\mathbb{R})$ *is given by*

$$\|A\|_F = \sqrt{\sum_{i,j\in[1,n]^2} |a_{ij}|^2}.$$

We remind a classical result related to the Frobenius norm.

Proposition 9 *Let us consider* $A,\ U \in M_n(\mathbb{R})^2$ *such as* $\|U\|_F = 1$*, then we have:*

$$\|A\,U\|_F = \|U\,A\|_F = \|A\|_F.$$

Going back to the MDS, our goal is to find $\mathcal{F}'$ that minimizes:

$$\Phi(\mathcal{F}') = \sum_{i,j\in[1,n]^2} (\|v'_i - v'_j\|^2 - \|v_i - v_j\|^2).$$

Let us call F the matrix associated to the dataset $\{v_i\}_{i\in[1,n]} \in \mathbb{R}^D$, F' the matrix associated to $\{v'_i\}_{i\in[1,n]} \in \mathbb{R}^d$, S and S' respectively the Euclidean squared distance matrices of F and F', and G and G' respectively their Gram matrices.

Another way to see the optimization problem of MDS is to seek for a matrix S' that minimizes:

$$\Phi(S') = \sum_{i,j\in[1,n]^2} (S'_{ij} - S_{ij}),\ \text{with rank}(S') = d, \tag{1.18}$$

such that

$$\sum_{i,j\in[1,n]^2} (S'_{ij} - S_{ij}) \leq \sum_{i,j\in[1,n]^2} |S'_{ij} - S_{ij}|.$$

Thanks to the norms equivalence relationship in $M_n(\mathbb{R})$, we have:

$$\exists \alpha \in \mathbb{R}^+,\ \|S' - S\|_1 = \sum_{i,j\in[1,n]^2} |S'_{ij} - S_{ij}| \leq \alpha \cdot \|S' - S\|_F$$

Therefore, our new goal is to find S' that minimizes:

$$\Phi(S') = \|S' - S\|_F,\ \text{with rank}(S') = d, \tag{1.19}$$

Then, using the proposition 9 with $\tilde{H} = (\|H\|_F)^{-1} \cdot H$, we have:

$$\|S' - S\|_F = \|\tilde{H}\,(S' - S)\,\tilde{H}\|_F = \frac{1}{\|H\|_F^2} \cdot \|H\,(S' - S) \times H\|_F,$$
$$\|S' - S\|_F = \frac{2}{\|H\|_F^2} \cdot \|G' - G\|_F.$$

Finally, we conclude that our problem is equivalent to find G' that minimizes:

$$\Phi(G') = \|G' - G\|_F,\ \text{with rank}(S') = d. \tag{1.20}$$

To do it, we use now the classical Eckart and Young theorem [41].

Theorem 10 *Let us consider a matrix $A \in M_{nm}(\mathbb{R})$, with $(n, m) \in \mathbb{N}^2$. We can write its singular value decomposition $A = U_A \Sigma_A V_A^T$, where $\Sigma_A = diag(\sigma_1, \ldots, \sigma_r)$ is diagonal and contains the singular value of A sorted in decreasing order, and $U_A \in M_{nr}(\mathbb{R})$ and $V_A \in M_{rm}(\mathbb{R})$ have orthogonal vectors that contain the left and right singular vector of A, and where $r = rank(A)$. Then if we call $A' \in M_{nm}(\mathbb{R})$ the best d-rank approximation of A under the Frobenius norm which solution to:*

$$\|A' - A\|_F, \text{ with } rank(S') = d$$

is $A' = U_{A,d} \Sigma_{A,d} V_{A,d}^T$, where $\Sigma_{A,d}$ is diagonal and contains the top d singular value of A sorted in decreasing order, and $U_A \in M_{nr}(\mathbb{R})$ and $V_A \in M_{rm}(\mathbb{R})$ contain respectively the associated left and right singular vector of A. Moreover the error of minimisation is given by:

$$\|A' - A\|_F = \sqrt{\sum_{i=d+1}^{r} \sigma_i^2}.$$

Then, applying this theorem to our problem, we have

$$G' = U_{G,d} \Sigma_{G,d} U_{G,d}^T, \tag{1.21}$$

where G' is squared, since $G' = (F')^T F'$, we finally have

$$F' = \Sigma_{G,d}^{1/2} U_{G,d}^T, \tag{1.22}$$

where $\Sigma_{G,d}$ is diagonal and contains the top d eigenvalue of G and $U_G \in M_n(\mathbb{R})$ contains the associated eigenvectors.

Isomap - A nonlinear method

Let us consider again a set of n points $F = \{v_i\}_{i=1}^n \in \mathbb{R}^D$. Isometric mapping (Isomap) [149] is a generalization of MDS, where the pairwise distance between points (v_i, v_j) is replaced by an estimation of the pairwise geodesic distance between points (v_i, v_j). The basic idea consists in building a weighted graph which goal is to recover the local geometry of the manifold, each node of this graph is a point v_i and each edge is an Euclidean distance between the nodes. Then, some edges are removed, in order to build local neighbourhoods. There are two ways to build the graph: either $k-$neighbourhood (i.e., the number of neighbours is fixed to k) or $\epsilon-$neighbourhood (i.e., a threshold of value ϵ is applied on the distance). Dijkstra algorithm is used to compute the shortest paths in the graph. It turns out that under certain conditions regarding the sampling of the manifold, its convexity and the fact that the manifold is smooth, the graph distance happens to be a good approximation of the geodesic distance on the manifold. Therefore the cost function to minimize in Isomap is

$$\Phi(\mathcal{F}') = \sum_{i,j \in [1,n]^2} (\|v_i' - v_j'\|^2 - d_{\text{geo}}(v_i, v_j)^2), \tag{1.23}$$

where d_{geo} is an approximation of the geodesic distance on the manifold.

Locally linear Embedding (LLE) - A nonlinear method

Locally Linear Embedding (LLE) is an unsupervised learning algorithm introduced by Roweis and Saul [127] that reduces the dimensionality of a dataset of points $\mathcal{F} = \{v_i\}_{i=1}^n \subset \mathbb{R}^D$ based on their local structure. It is based on the intuition that each initial point $\forall i \in [1, n]$ v_i can be linearly reconstructed by its neighbours ($k-$neighbourhood or $\epsilon-$neighbourhood). The error of reconstruction is measured given by:

$$\Phi_1(W) = \sum_i \|v_i - \sum_j W_{ij} \cdot v_j\|^2,$$

where W_{ij} represents the contribution of the jth coordinate over v_i, with the constrain that $\sum_j W_{ij} = 1$ and $W_{ij} = 0$ if v_j does not belong to the neighbourhood of v_i. This constraint, together with the cost function, makes that the reconstructed data are invariant to rotations, rescaling, and translation. Then one transforms v_i to v_i' thanks to a mapping consisting in translation, rotation, and rescaling. Since W_{ij} are invariant to this mapping, they are also the weight of reconstruction of v_i'. Thus to find v_i', we just need to minimize the following equivalent cost function:

$$\Phi_2(F') = \sum_i \|v_i' - \sum_j W_{ij} \cdot v_j'\|^2, \tag{1.24}$$

with the constrain $\sum_j F'_{ij} = 0$, $\forall j \in [1, n]$ and $F'^T F' = I$.

Hessian Eigenmaps - A nonlinear method

Hessian Eigenmaps introduced by Donoho and Grimes [39] achieves an embedding by minimizing the Hessian functional estimated from the sampled manifold. One of the major advantages of Hessian Eigenmaps is that it can recover the true manifold geometry even if it is not convex. Let us consider a set of n points $F = \{v_i\}_{i=1}^n \in \mathbb{R}^D$ that lie on a manifold $\mathcal{F}$ of intrinsic dimension d. For any point v_i we have a local orthonormal coordinate system $(v_i^1, \ldots, v_i^d)$ on the tangent space $\mathbb{T}_{v_i}(\mathcal{F})$. Let us consider the projection operator $\mathrm{Pr}_{\mathbb{T}_{v_i}}(v_j)$, where v_j is in a neighbourhood of v_i. For any twice differentiable function $g : \mathcal{F} \longrightarrow \mathbb{R}$ it is possible to define its tangent Hessian matrix as:

$$H_g^{\tan}(v_i)_{k,l} = \frac{\partial}{\partial v_i^k} \frac{\partial}{\partial v_i^l} g(\mathrm{Pr}^{-1}_{\mathbb{T}_{v_i}}(v_i)).$$

Let us define the quadratic form $\mathcal{H}(g) = \int_{\mathcal{F}} \|H_g^{\tan}(v_i)\|_F^2 dv$ which represents the average over the data manifold $\mathcal{F}$ of the Frobenius norm of the Hessian matrix of g. Donoho and Grimes [39] proposed and proved the following key result: $\mathcal{H}(g)$ has a $d+1$ dimensional null-space which consists in the constant function and the d-dimensional function spanned by the isometric coordinates. From this result, it is possible to formulate an algorithm to approximate the computation of Hessian Eigenmaps.

1.3.3 Kernels and Kernel PCA - A nonlinear method

Kernel trick

Kernel methods are used to define nonlinear decision functions while using linear methods as basic ingredients. They allow to project the data into a nonlinear space

that can be even of infinite dimension. For more information on kernels the reader is referred to [133].

Definition 11 *A kernel is a function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ *which is symmetric and hermitian.*

However most of the time, we work with positive definite kernels.

Definition 12 k *is called a positive definite kernel if* $\forall\{x_1, \ldots, x_n\} \in \mathcal{X}^n$ *and* $\forall\{\alpha_1, \ldots, \alpha_n\} \in \mathbb{C}^n$*, the following non-negativity condition holds:*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j^* k(x_i, x_j) \geq 0.$$

If we consider a dataset $\{x_1, \ldots, x_n\} \in \mathcal{X}^n$, thanks to positive definite kernel, it is possible to define a similarity matrix:

$$[K]_{ij} := k(x_i, x_j),$$

which is positive semi-definite (PSD). Usually this matrix is also called the Gram matrix, or the kernel matrix.

Positive definite kernels are considered as a generalization of inner product on nonlinear spaces, and so can be used to replace the inner product in algorithms. In addition, the function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, defined by: $\forall(i, j) \in [1, n]$ $k(x_i, x_j) =< x_i, x_j >_{\mathbb{R}^D}$, where $< .., .. >_{\mathbb{R}^D}$ is the inner product on $\mathbb{R}^D$, is just called the linear kernel.

Definition 13 *A Hilbert space* $\mathcal{H}$ *is a vector space with a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product. That means that every Cauchy sequence in* $\mathcal{H}$ *has a limit in* $\mathcal{H}$*.*

We remind now the important Moore-Aronszajn theorem for kernel methods.

Theorem 14 k *is a positive definite kernel if and only if there exists a Hilbert space* $\mathcal{H}$ *and a mapping* $\phi : \mathcal{X} \to \mathcal{H}$ *such that* $k(x_i, x_j) =< \phi(x_i), \phi(x_j) >_{\mathcal{H}}$*.*

Definition 15 *Let* $(\mathcal{H}, < .., .. >_{\mathcal{H}})$ *be a Hilbert space consisting of functions of* $\mathcal{X}$ *in* $\mathbb{C}$*. The function* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ *is the reproducing kernel of* $\mathcal{H}$*, provided that the latter admits one if and only if:*

- *For any element* $x \in \mathcal{X}$*, the function* $k(x, .) : t \to k(x, t)$ *belongs to* $\mathcal{H}$*;*
- *For all* $x \in \mathcal{X}$*,* $g \in \mathcal{H}$*, the reproducing property is verified:* $g(x) =< g, k(x, .) >_{\mathcal{H}}$*. If a reproducing kernel exists, then* $\mathcal{H}$ *is called a reproducing kernel Hilbert space (RKHS).*

It can be shown that every positive definite kernel is the reproducing kernel of at most one unique RKHS of functions from $\mathcal{X}$ to $\mathbb{C}$. Reciprocally, if $\mathcal{H}$ is a RKHS, then it has a unique reproducing kernel. Moreover every reproducing kernel is a positive definite kernel. A proof of these theorems may be found in [122]. From these properties, it results that we can replace $\phi(x_i)$ by $k(x_i, .)$ and therefore to obtain:

$$k(x_i, x_j) =< \phi(x_i), \phi(x_j) >_{\mathcal{H}} =< k(x_i, .)), k(x_j, .) >_{\mathcal{H}} .$$

A final key result, named the Representer theorem.

Theorem 16 *Let $\mathcal{X}$ be a set endowed with a positive definite kernel k, and $\mathcal{H}_k$ the corresponding RKHS, and $x_1, \ldots, x_n \subset \mathcal{X}$ a finite set of points. Let $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ be a function of $n + 1$ variables, strictly increasing with respect to the last variable. Then, any solution to the optimization problem:*

$$\min_{f \in \mathcal{H}_k} \Psi(g(x_1), \ldots, g(x_n), \|g\|_{\mathcal{H}_k}), \tag{1.25}$$

admits a representation of the form:

$$\forall x \in \mathcal{X},\ g(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x), \tag{1.26}$$

where $\|g\|_{\mathcal{H}_k} = \sqrt{< g, g >_{\mathcal{H}_k}}$.

The concept of "kernel trick" corresponds to the idea that for any algorithm that uses only the inner product between the input vectors, we can do this implicitly, in a Hilbert space called feature space ϕ, by replacing each inner product by an evaluation of the kernel. This evaluation can be done without having a representation of the feature map. The kernel trick allows to operate in a space larger without having to explicitly calculate the coordinates of the data in this space.

Kernel PCA

Let us consider a set of n vectors $v_i \in \mathbb{R}^D$, $\forall i \in [1, n]$. Let us map our data onto another space $\mathcal{H}$, that have some interesting properties, by the mapping:

$$\phi = \begin{cases} \mathbb{R}^D \to \mathcal{H} \\ v_i \to \phi(v_i) \end{cases} \tag{1.27}$$

where ϕ is a function that may be nonlinear, and depends on the kernel. The choice of the kernel is a difficult question that will be discussed latter. Now let us suppose that we have chosen a kernel.

The goal of the kernel PCA (KPCA) is to find the set of vectors $\{w_j, j \in [1, D]\}$ that minimize the following cost function:

$$\min \left(\frac{1}{n} \sum_{i}^{n} \|\phi(v_i) - < \phi(v_i), w_j >_{\mathcal{H}_k} \cdot w_j\|^2_{\mathcal{H}_k} \right),\ \forall j \in [1, P], \tag{1.28}$$

where $w_j,\ j \in [1, n]$ are the so-called principal components. By developing the distance one gets:

$$\begin{aligned} \|\phi(x_i) - < \phi(v_i), w_j >_{\mathcal{H}_k} \cdot w_j\|^2_{\mathcal{H}_k} = \\ \|\phi(v_i)\|^2_{\mathcal{H}_k} - 2 < \phi(v_i), w_j >^2_{\mathcal{H}_k} + < \phi(v_i), w_j >^2_{\mathcal{H}_k} \|w_j\|^2_{\mathcal{H}_k} \end{aligned} \tag{1.29}$$

By adding the constraint that $\|w_j\|^2_{\mathcal{H}_k} = 1$, replacing (1.29) in 1.28 and keeping the term that depend on w_j, we have the following new objective function:

$$\max(\frac{1}{n} \times \sum_{i}^{n} < \phi(v_i), w_j >^2_{\mathcal{H}_k}),\ \text{with } \|w_j\|^2_{\mathcal{H}_k} = 1,\ \forall j \in [1, P]. \tag{1.30}$$

Problem (1.30) can be rewritten thanks to the Lagrange multiplier theorem as follows

$$\mathbf{L}(w_j, \lambda) = \frac{1}{n} \sum_{i}^{n} < \phi(v_i), w_j >^2_{\mathcal{H}_k} -\lambda \left(\|w_j\|^2_{\mathcal{H}_k} - 1 \right) \tag{1.31}$$

where $\lambda \in \mathbb{R}$. Thanks to the Representer theorem presented above, w_j can be written as:

$$w_j = \sum_{l=1}^{n} \alpha_{l,j} \phi(v_l). \tag{1.32}$$

Therefore, finally, we have:

$$\mathbf{L}(\alpha_j, \lambda) = \frac{1}{n} \sum_{i}^{n} \left(\sum_{l=1}^{n} \alpha_{l,j} < \phi(v_i), \phi(v_l) >_{\mathcal{H}_k} \right)^2 - \lambda \left(\sum_{l=1}^{n} \sum_{t=1}^{n} \alpha_{l,j} \alpha_{t,j} k(v_l, v_t) - 1 \right),$$

which can be rewriten in a matrix way as:

$$\mathbf{L}(\alpha_j, \lambda) = \frac{1}{n} \alpha_j^t \, k^2 \, \alpha_j - \lambda \cdot \left(\alpha_j^t \, k \, \alpha_j - 1 \right). \tag{1.33}$$

By deriving it, we obtain:

$$\frac{\partial \mathbf{L}}{\partial \alpha_j}(\alpha_j, \lambda) = \frac{2}{n} k^2 \, \alpha_j - 2\lambda \, k \, \alpha_j = 0. \tag{1.34}$$

In conclusion, KPCA optimizaton problem is equivalent to eigenproblem:

$$k \, \alpha_j = n\lambda \cdot \alpha_j \tag{1.35}$$

such that (1.34) and (1.35) are equivalent for non null eigenvalue, in the other cases, the solutions are not interesting for the maximization problem, and would not be taken into consideration.

Finally α_j, $j \in [1, D]$, are the eigenvectors of k, and the normalization impose that the eigenvector are:

$$\tilde{\alpha_j} = \frac{1}{\lambda_j} \alpha_j.$$

Importance of the choice of the kernel

The advantage of the kernel trick is that we can use many different kernels without having to compute explicitly the mapping $\phi(v_i)$. Two of the most popular kernels are:

- The polynomial kernel:

$$k(v_i, v_j) = (< v_i, v_j >_{\mathbb{R}^D} + c)^P,$$

  where P is the degree of the kernel and c a scalar constant;

- The radial basis function (rbf) kernel, often called Gaussian kernel:

$$k(v_i, v_j) = e^{\frac{-\|v_i - v_j\|^2_{\mathbb{R}^D}}{2\sigma^2}},$$

  with scaling parameter σ. This kernel bring the data into a RKHS of infinite dimension.

Kernel function $k(v_i, v_j)$ can be seen as an interaction function between v_i and v_j, that quantifies the similarity between two objects. Here we work on high dimensional spaces, where the notion of similarity is a key issue to address the curse of dimensionality. Usually we want a good similarity measure to be high for data that belong to the same cluster and small on the contrary. However since we do not have access to this information, this involves that the question of similarity in high dimensional spaces is far from being simple.

We have represented in Figures 1.5 and 1.6 the results of KPCA for different kernels. These results illustrate the importance of the choice of the kernel. By the way, the choice of σ in the case of the rbf kernel has a strong impact on the dimensionality reduction, as shown in the figures.

In this thesis, and when a particular approach is not mentioned, we have chosen for the rbf kernel parameter

$$\sigma = \text{median}(\{\|v_i - v_j\|_{\mathbb{R}^D},\ (i,j) \in [1,n]^2\}).$$

For readers interested on the choice of σ are referred to [148].

1.3.4 Dictionary learning

The goal of dictionary learning is to find a sparse representation which approximates an image from a set of consistent images. Let us consider that we have a collection of D images, denoted here by y_j, with $j \in [1, D]$. Our goal is to find a dictionary $\mathcal{D}$, composed of d atoms ϕ_k, with $k \in [1, d]$, such as each image y_j can be expressed as a linear combination of atoms from the dictionary:

$$y_j = \sum_k \alpha_{j,k}\phi_k = \alpha_j^T \phi, \quad \forall j. \tag{1.36}$$

In order to find a unique dictionary, some constraints of sparsity are imposed on α_j. Typically the objective is to find a sparse vector α_j that would contain a small number of non-zero coefficients. The reformulated optimization problem is written as follows [152]:

$$\min \left[\underbrace{\|y_j - \alpha_j^T \phi\|}_{\text{Term of estimation}} + \lambda \underbrace{\|\alpha_j\|_0}_{\text{Term of sparseness}} \right], \ \forall j \in [1, D]. \tag{1.37}$$

In order to solve this problem, the proposed techniques can be divided into two families. First, supervised techniques, where the dictionary $\mathcal{D}$ is given, and so the goal is only to find α_j. We do not consider this kind of approaches. Second, unsupervised techniques, where $\mathcal{D}$ and α_j are both estimated, which can be subdivided into three main kinds: i) probabilistic methods, ii) clustering-based methods, and iii) learning a dictionary with specific structure.

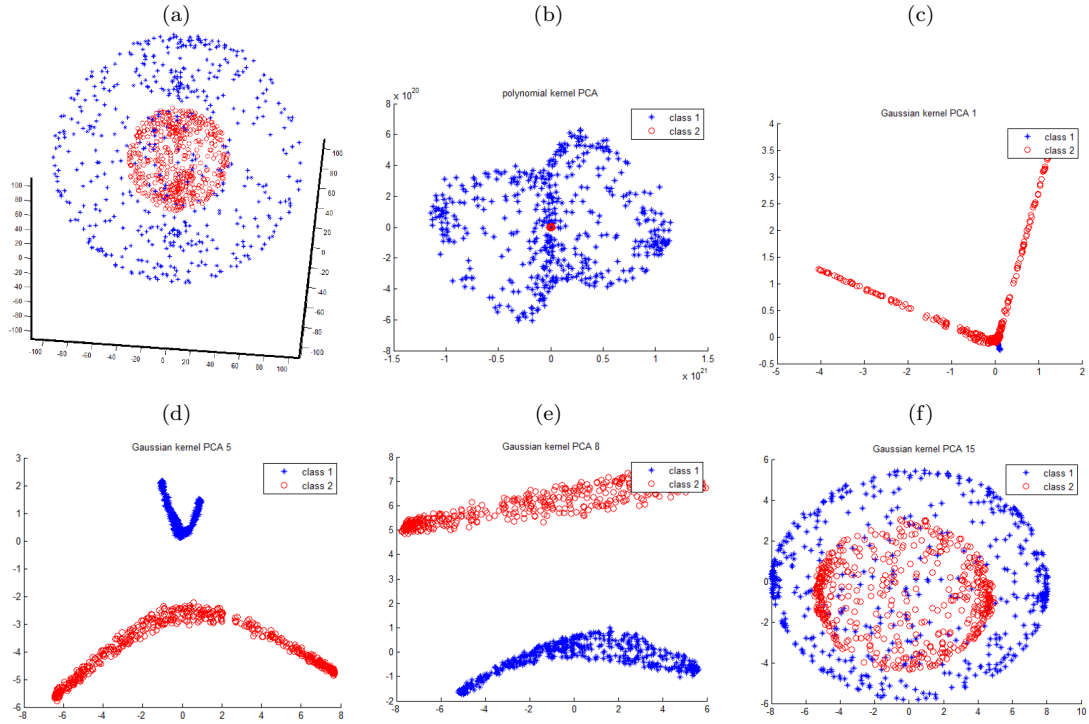


Figure 1.5: (a) Synthetic manifold with two concentric spheres, (b) polynomial KPCA with $p = 5$, (c) Gaussian KPCA with σ, (d) Gaussian KPCA with 5σ,(e) Gaussian KPCA with 8σ, (f) Gaussian KPCA with 15σ.

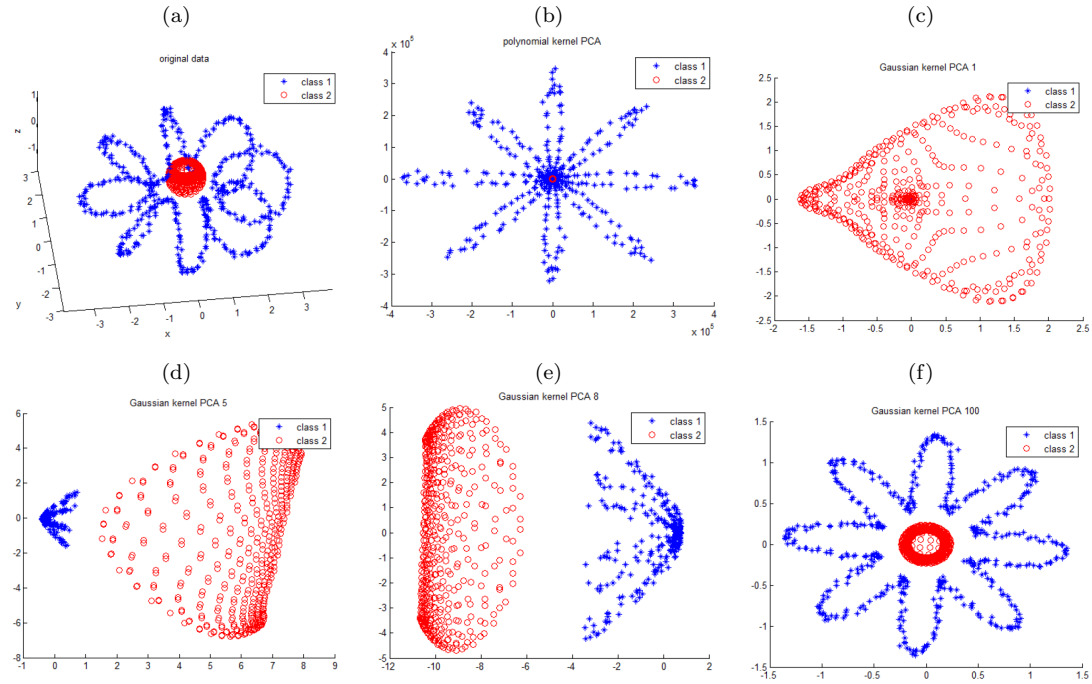


Figure 1.6: (a) Flower-like synthetic manifold, (b) Polynomial KPCA with $p = 5$, (c) Gaussian KPCA with σ, (d) Gaussian KPCA with 5σ,(e) Gaussian KPCA with 8σ, (f) Gaussian KPCA with 100σ.

Maximum likelihood method - A probabilistic method

The maximum likelihood method for dictionary learning [95, 82] leans on the hypothesis that the signals can be written as:

$$y_j = \alpha_j^T \phi + n_j, \quad \forall j,$$

where n_j is a white Gaussian residual vector with covariance σ^2. Then the goal of the approach is to find α and $\mathcal{D}$ maximizing the likelihood $P(Y|\mathcal{D}) = \Pi_{i=1}^{D} P(y_i|\mathcal{D})$ with:

$$P(y_i|\mathcal{D}) = \int P(y_i, \alpha|\mathcal{D}) d\alpha = \int P(y_i|\alpha, \mathcal{D}) P(\alpha) d\alpha.$$

VQ objective - A clustering based method

In this technique [130], where the datapoints are partitioned into K clusters with for instance K-means algorithm, each data cluster is approximated thanks to one atom of the dictionary, which usually corresponds to the central point (i.e., mean) of the cluster.

The K-SVD - A clustering based method

K-SVD [4] is a powerfull generalization of VQ objective. It is contrary to most of the methods where the optimisation problem (1.37) is solved thanks to a double optimization problem, where one first fix α, and find $\mathcal{D}$, and then, one fixes $\mathcal{D}$, to find α. In K-SVD each atom ϕ_k of $\mathcal{D}$ is update sequentially using an Singular Value Decomposition (SVD), then thanks to the SVD, a new atom $\tilde{\phi_k}$ is found as well as its corresonding coefficients. This technique is in a way a generalisation of the VQ objective since each data y_j can be expressed thanks to multiple atoms, from multiple partitions.

Unions of orthonormal Bases - Structured dictionary

The technique [81] is based on a particular constrained optimization problem, considering that the solution is a set of union of orthonormal bases (1.37). Its rationale is founded on the fact that many signals can be seen as a set of orthonormal bases, and also that, thanks to this structure, the problem of optimization is easier to solve. In this method, each bases is updated sequentially, but the optimization problem does not update the atom of the basis and the corresponding coefficients at the same time.

1.4 State-of-the-art on classification

1.4.1 Regression

Supervised classification can be seen as a special case of regression. For this reason, let us start by introducing a background on regression [16].

Let us suppose that we have n datapoints (or vectors) $\{v_i\} \in \mathbb{R}^D$, with $i \in [1, n]$. Each of these points is associated to a target value $\{t_i\} \in \mathbb{R}$. The set $\{v_i, t_i\}$, $i \in [1, n]$, is called the training set. The goal of regression is to predict the target

value t_i for a new data v_i with $i \notin [1, n]$. To be able to predict such a value t_i, we can consider that it can be written as a linear combination of the coordinates of v_i. Let us write $v_i = (v_{i,1}, \ldots, v_{i,j}, \ldots, v_{i,D})^t \in \mathbb{R}^D$. A simple model often used in regression is to consider that the prediction function is given by:

$$y(\omega, v_i) = \omega_0 + \omega_1 v_{i,1} + \ldots + \omega_D v_{i,D} = \omega_0 + \sum_{j=1}^{D} \omega_j v_{i,j}. \tag{1.38}$$

Our goal is to learn the parameters $\{\omega_0, \ldots, \omega_D\}$ thanks to the training set. This model (1.38) is called linear regression, and may have some limitations. That is why we prefer to consider a more general model:

$$y(\omega, v_i) = \omega_0 + \sum_{j=1}^{M} \omega_j \phi_j(v_i), \tag{1.39}$$

where $\phi_j(v_i)$ represents a set of basis functions, with M the number of basis. There are many choices of basis functions. For instance, we can use a polynomial basis

$$\phi_j(v_i) = v_{i,j}^k, \tag{1.40}$$

where the power k is a hyper-parameter or we can consider the rbf:

$$\phi_j(v_i) = \exp\left(-\frac{\|v_i - \mu_j\|^2}{2\sigma^2}\right), \tag{1.41}$$

where now the hyper-parameters are μ_j, which governs the spatial locations, and σ the scaling factor.

Let us consider that the target data is given by the previous deterministic function, corrupted by Gaussian noise ϵ of zero mean Gaussian and inverse variance β, such that:

$$t_i = y(\omega, v_i) + \epsilon,$$

where $y(\omega, v_i) = \omega^t \phi(v_i)$, with $\omega^t = (\omega_0, \ldots, \omega_M)$, $\phi(v_i) = (\phi_0(v_i), \ldots, \phi_M(v_i))^t$ and $\phi_0(v_i) = 1$. Let us denote $\mathcal{E}$ the random variable associated to the noise such that $\mathcal{E} \sim \mathcal{N}(0, \beta^{-1})$. We can write τ for the random variable associated to the target value, such that we have $\tau \sim \mathcal{N}(y(\omega, v_i), \beta^{-1})$, which depends on two parameters, ω and β and a spatial position v_i on the manifold of the data. One can see that this is related to the notion a random functions that will be introduce in the sequel of the thesis. Let us consider that the training set is drawn independently from the previous law. Then we can write the likelihood function of the parameters ω and β:

$$\mathcal{L}(t_1, \ldots, t_n/\omega, \beta) = \prod_{i=1}^{n} \frac{\sqrt{\beta}}{\sqrt{2\pi}} \exp\left(\frac{-\beta(t_i - y((\omega, v_i))^2}{2}\right).$$

Taking the logarithm of the likelihood function, we have:

$$\log \mathcal{L}\left(t_1, \ldots, t_n/\omega, \beta\right) = \sum_{i=1}^{n} \left(1/2. \log \beta - 1/2 \log 2\pi - \beta/2(t_i - y((\omega, v_i))^2\right).$$

If we want to find the set of parameters that maximize the likelihood, we have first to derive it according to each of the parameters of the log-likelihood, and equals it to zero. On the previous expression the term that depends just on ω is:

$$E_d(\omega) = \frac{\beta}{2}\sum_{i=1}^{n}(t_i - y((\omega, v_i))^2.$$

Our goal is to find ω that minimizes E_d. One can recognize the typical cost function of machine learning regression. In order to minimize this cost function, we can derive it, equal it to zero, to finally obtain that:

$$\omega_{ML} = (\Phi^t\,\Phi)^{-1}\,\Phi^t\,t, \tag{1.42}$$

where $\Phi \in M_{n,M+1}(\mathbb{R})$ is defined by

$$\Phi = \begin{pmatrix} \phi_0(v_1) & \dots & \phi_M(v_1) \\ \vdots & \ddots & \vdots \\ \phi_0(v_n) & \dots & \phi_M(v_n) \end{pmatrix} \tag{1.43}$$

and where $t = (t_1, \dots, t_n)^t$ is the vector of all the training targets values. It is also possible to estimate β_{ML} as:

$$\beta_{ML} = \frac{1}{n}\sum_{i=1}^{n}\left(t_n - \omega_{ML}^t\phi(x_i)\right)^2, \tag{1.44}$$

such that β_{ML} provides us information on the precision of the regression.

This is the classical case of regression. However if one has access to some prior knowledge on ω, this can help to regularize the cost function. Let us consider that the random variable associated with ω follows a normal distribution $\mathcal{N}(m_0, S_0)$, which is called the prior distribution. Then we need to compute the posterior distribution and we can estimate the new value of ω. As shown in [16] (page 153), in the case where $m_0 = 0 \in \mathbb{R}^{M+1}$ and $S_0 = \alpha \cdot I$, with I the identity matrix of size $M+1$, then the cost function is called ridge regression is just given by:

$$E_B(\omega) = \frac{\beta}{2n}\sum_{i=1}^{n}(t_i - y((\omega, v_i))^2 + \alpha \cdot \omega^t\,\omega, \tag{1.45}$$

whose solution is equal to:

$$\omega_B = \left(\Phi^t\,\Phi + \frac{\alpha}{\beta}\cdot I\right)^{-1}\,\Phi^t t \tag{1.46}$$

where this new term $\alpha/\beta \cdot I$ can be seen as a way to improve the inversion of the matrix, since small eigenvalues that might be problematic are improved thanks to this term. In addition, in the regularization theory [55, 133], it can also be seen as a way to avoid overfitting on the training set.

In the previous case, we were looking for a $\omega \in \mathbb{R}^{M+1}$ of finite dimension. By means of the kernel trick, it is now possible to have a potential infinite dimension ω. Let us map the training set $v_i \in \mathbb{R}^D$, $\forall i \in [1, n]$, onto another space using

$$\phi = \begin{cases} \mathbb{R}^D \to \mathcal{H} \\ x_i \mapsto \phi(x_i) \end{cases} \tag{1.47}$$

where again, ϕ is a function that may be nonlinear, and depends on the choice of the kernel. Then, in our regression case, the cost function can be now reformulated as:

$$E_B(\omega) = \frac{1}{2n}\sum_{i=1}^{n}(t_i - y(\omega, v_i))^2 + \lambda \cdot \omega^t\,\omega. \tag{1.48}$$

Apparently nothing is different, however since $\mathcal{H}$ might be infinite dimensional, and since we might not have access to ϕ, then the problem has changed. In order to solve this problem, we use the Representer theorem, and say that a solution of the problem must be of the form:

$$y(\omega, v_p) = \sum_{i=1}^{n}\gamma_i\, k(v_p, v_i), \tag{1.49}$$

where k is the kernel associated to ϕ, and where v_p is not necessarily in the training set. At first v_p would be in the training set to learn the parameters γ_i, then it will not be. We can rewrite this problem of regression in a matrix way:

$$E_B(\gamma) = \frac{1}{2n}\left(K\gamma - t\right)^t\left(K\gamma - t\right) + \lambda \cdot \gamma^t K\gamma, \tag{1.50}$$

where $\gamma = (\gamma_1, \ldots, \gamma_n)$ and K is the Gram matrix associated with the kernel such that $K_{ij} = k(v_i, v_j)$. The expression (1.50) can be developped as follows:

$$E_B(\gamma) = \frac{1}{2n}\left(\gamma^t K^2\gamma - 2t^t K\gamma + t^t t\right) + \lambda \cdot \gamma^t K\gamma.$$

Then, by deriving it and setting equal to zero we obtain:

$$\gamma = (K + \lambda \cdot I)^{-1}t, \tag{1.51}$$

and finally, the solution is:

$$y(\omega, v_p) = \gamma^t k(v_p, :), \tag{1.52}$$

with $k(v_p, :) = (k(v_p, v_1), \ldots, k(v_p, v_n))^t$. It happens that in the case $\lambda = 0$, the solution of the kernel regression is the same as the one of the kriging interpolation without constrain in the coefficients of the linear combination, see corresponding chapter.

1.4.2 Classification

From regression to classification

The problem of classification can be considered as a special case of regression, where the target value is assigned to one of the K discrete classes. Then the input space is divided into regions, where each region is assigned to one class. Thus the goal of classification algorithms is to find the decision boundaries between the classes.

In classification, the target vector t_i of for instance class 2 is typically written as:

$$t_i = (0, 1, 0, \ldots, 0)\,. \tag{1.53}$$

As discussed above, in linear regression, the model is of the form: $y(v_i) = \omega_0 + \sum_{j=1}^{D} \omega_j v_{i,j}$. It turns out that in classification, since most of the time we want that the result of y is in $[0, 1]$, it is possible to use an "activation function" a such that the model becomes:

$$y(v_i) = a\left(\omega_0 + \sum_{j=1}^{D} \omega_j v_{i,j}\right). \tag{1.54}$$

From binary hyperplane classification to multi-class hyperplane classification

Binary classification At first, we address the simple case of binary classification. We denote these two classes $t_i \in \{-1, +1\}$. In addition, we consider that

$$y(v_i) = \omega_0 + \sum_{j=1}^{D} \omega_j v_{i,j} = \omega_0 + \omega^t v_i.$$

In the rest of the chapter, we write $b_0 = \omega_0$, since it represents a bias. We look for the weights ω such that if the data point v_i is of the class 1 then $y(v_i) \geq 0$, and $y(v_i) < 0$ if the point is of class -1. Hence, this activation function is just $\text{sign}(x)$, which gives the sign of x, thus it separates the space in two classes.

Let us consider two points v_1 and v_2 which are on the decision boundary, therefore $y(v_1) = y(v_2) = 0$. Thus $\omega^t(v_1 - v_2) = 0$, then ω is orthogonal to the decision boundary. Finding ω will therefore determine the direction of the decision boundary. Moreover if v_1 is a vector on the decision boundary, $y(v_1) = 0$, so

$$\frac{\omega^t v_1}{\|\omega\|} = -\frac{b_0}{\|\omega\|}.$$

This implies that the distance from the origin to the decision surface is determined by ω_0. We illustrate this property in Figure 1.7, for a case of $D = 2$. In addition, for any point v_i, we can decompose it into the base composed of the information tangent to the decision boundary $v_i^{\perp}$ and the information tangent to the w, i.e.,

$$v_i = v_i^{\perp} + r\frac{\omega}{\|\omega\|},$$

where $r \in \mathbb{R}$ represents a coordinate. Then we have on the one hand:

$$\omega^t v_i = \omega^t v_i^{\perp} + \ r\frac{\omega}{\|\omega\|},$$

and on the other hand, we add ω_0 and use the fact that $y(v_i^{\perp}) = 0$, to obtain that

$$y(v_i) = r\frac{\omega}{\|\omega\|}. \tag{1.55}$$

Instead of considering the classification in the linear space, one might work also on the kernel space. To do that, we just need a feature space ϕ and then to apply the classification on this space thanks to

$$y(v_i) = \omega_0 + \omega^t \phi(v_i).$$

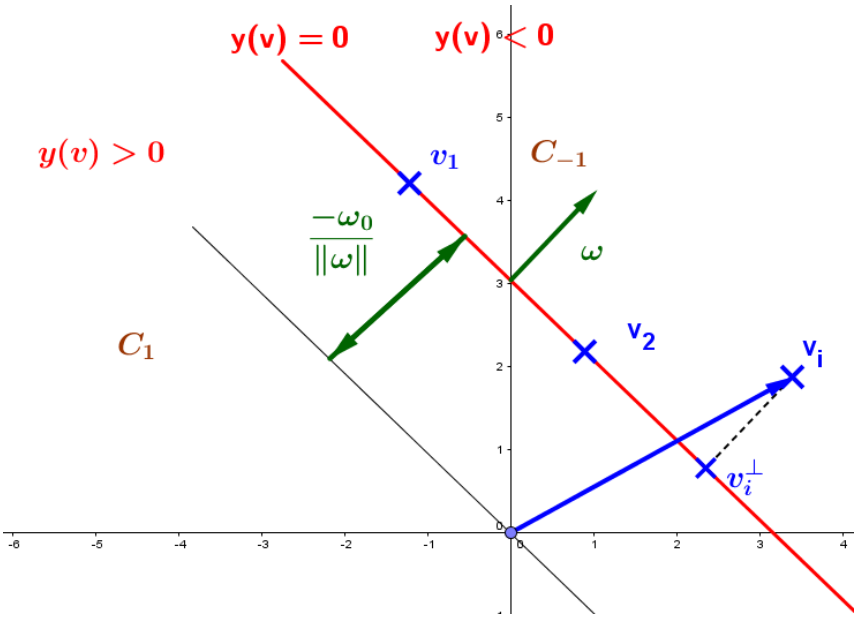


Figure 1.7: Illustration of the geometry of the hyperplane for binary classification.

Multiclass classification Let us now consider the problem of multiclass classification. To transform a problem of binary classification into one of multiclass classification, there are different approaches that can be applied. The first one consists in proceeding to multiple binary classification between one class and all the others. This paradigm of multiclass classification is called one-versus-all. By doing that, we evaluate for each point the probability of being in a given class in with respect to the rest of the classes. This technique presents some uncertainties as illustrated in the example Figure 1.8(a). Another possibility is to proceed to multiple binary classification of one class against another one. This kind of approaches is called the one-versus-one multiclass classification and it also presents some uncertainties, see Figure 1.8(b).

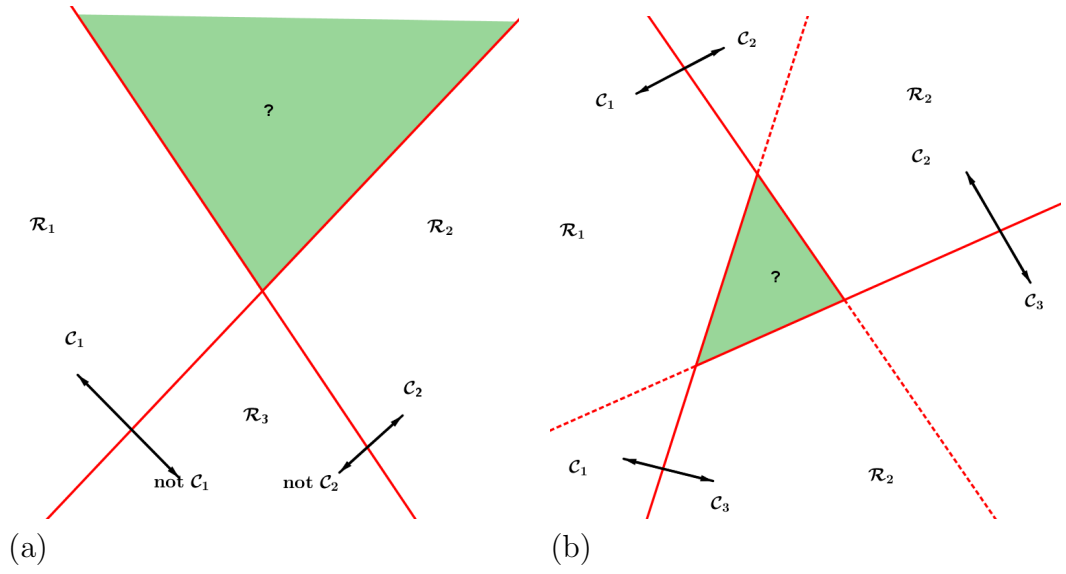


Figure 1.8: Illustration of the geometry of K-class classification from a set of 2-class classification techniques.

These issues can be avoided by considering a single K-class classification problem, and then to attribute to each target:

$$y(v_i)_k = \omega_k^t v_i + b_{0,k}, \tag{1.56}$$

such that each point is assigned to the class k such that $y(v_i)_k > y(v_i)_j$, for all $j \in [1, K]$. For more details about this technique see [16]. More recent techniques have been invented that provide interesting results, see for instance [31]. We conclude this paragraph by saying that multiclass classification is a rather difficult; open problem; since there are many alternative algorithms to transform from binary to multiclass classification.

Support Vector Machine (SVM)

We have now all the ingredients required to understand the Support Vector Machine (SVM) classification procedure. We start with the same binary classification problem: $y(v_i) = b_0 + \omega^t \phi(v_i)$. Since the weights ω might be an infinite dimensional vector, we change previous notation to the kernel one: $y(v_i) = g(v_i)$, with g being a function of the Hilbert space $\mathcal{H}$ associated with ϕ.

Since the distance of the data according to the hyperplane determines their classes, we might be interested in finding a hyperplane that is far away from the data. The corresponding distance is called the margin. According to the discussion of previous paragraphs, the distance of each point to the hyperplane is in binary classification given by

$$\frac{t_i g(v_i)}{\|g\|_{\mathcal{H}}}. \tag{1.57}$$

Hence the solution to find the maximum marging to the closest point v_i is the decision boundary, which can be found by solving:

$$\arg\max_{g \in \mathcal{H}} \left\{ \min_i \frac{g(v_i)}{\|g\|_{\mathcal{H}}} \right\}. \tag{1.58}$$

The intuition behind this idea is that for the definition of the hyperplane we focus just on the vectors which are near the boundary of influence. This can be formulated thanks to the hinge loss function, i.e.,

$$\Phi(t_i, y(g, v_i)) = \max\left(0, 1 - t_i g(v_i)\right). \tag{1.59}$$

This function is equal to zero if the data are on the correct side of the hyperplane. For data on the wrong side of the decision boundary, the value of the function is proportional to the distance from the margin. Thus the empirical risk function to be minimized is:

$$\min_{f \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n} \Phi(t_i, y(g, v_i)) \right] + \lambda \|g\|_{\mathcal{H}}, \tag{1.60}$$

where the parameter λ determines the trade-off between increasing the margin-size and ensuring that the v_i lie on the correct side of the margin, and it help also to regularize the solution.

Using one more time the Representer theorem, the solution of the previous problem can be expanded as:

$$g(v_p) = \sum_{i=1}^{n} \alpha_i k(v_i, v_p). \tag{1.61}$$

By plugging this formulation into the original problem, the following convex optimization problem is obtained:

$$\min_{\alpha \in \mathbb{R}^n} \left[\frac{1}{n} \sum_{i=1}^{n} \Phi(t_i \sum_{j=1}^{n} \alpha_i k(v_i, v_j)) \right] + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(v_i, v_j). \tag{1.62}$$

Different loss functions provide different classification algorithms. Here we use the hinge function. However, this function is not differentiable, so one should reformulate the problem as follows:

$$\min_{\alpha \in \mathbb{R}^n, \xi \in \mathbb{R}^n} \left[\frac{1}{n} \sum_{i=1}^{n} \xi_i \right] + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(v_i, v_j), \tag{1.63}$$

subject to:

$$\begin{cases} t_i \sum_{j=1}^{n} \alpha_i k(v_i, v_j) + \xi_i - 1 \geq 0 \\ \xi_i \geq 0 \end{cases} \tag{1.64}$$

The Lagrangian of this problem is given

$$\begin{aligned} \mathcal{L}(\alpha, \xi, \mu, \nu) = & \left[\frac{1}{n} \sum_{i=1}^{n} \xi_i \right] + \lambda \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j k(v_i, v_j) \\ & + \sum_{i=1}^{n} \mu_i (t_i \sum_{j=1}^{n} \alpha_i k(v_i, v_j) + \xi_i - 1) + \sum_{i=1}^{n} \nu_i \xi_i, \end{aligned}$$

which corresponds to a classical minimization of a convex quadratic function with linear constraints that can be solved using a quadratic program optimization package.

Neural network classification

The architecture (Artificial) neural networks are a classification approach which attempt to find a mathematical representation of how our biological system processes information. First, let us focus on the model. In classification, the optimization problem was modelled by:

$$y(v_i) = a(\omega_0 + \sum_{j=1}^{D} \omega_j \phi_j(v_i)), \tag{1.65}$$

where ϕ is representing a feature space associated with a kernel k, where the kernel is fixed and not learned. Here the goal is to make the kernel depending on many parameters which provide a huge flexibility to the feature space. Moreover we would

like that these parameters and, so the feature space, are learned during the classification process. Therefore, the classification will learn ω and ϕ. To construct the neural network, we first consider a linear model that depends on the input data:

$$c_k = \omega_{0,k}^{(1)} + \sum_{j=1}^{D} \omega_{j,k}^{(1)} v_{i,j}. \tag{1.66}$$

with $k \in [1, K]$ and where each c_k is a neurone of the first layer. The superscript (1) indicates that these parameters are the parameters of the first layer. Then, a nonlinear activation function a is applied on these quantities c_k:

$$z_k = a^{(1)}(c_k). \tag{1.67}$$

We can choose different kinds of activation functions, typically:

- A sigmoid function: $a = \tanh$; or
- Rectified Linear Unit (ReLU): $a(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$.

On the output of the first layer, a second linear combination is applied:

$$d_{k1} = \omega_{0,k1}^{(2)} + \sum_{k=1}^{K} \omega_{k,k1}^{(2)} z_k. \tag{1.68}$$

with $k1 \in [1, K1]$, which is followed by another activation function. We can combine the different stages and obtain:

$$y_{k1}(v_i) = a^{(2)} \left(\omega_{0,k1}^{(2)} + \sum_{k=1}^{K} \omega_{k,k1}^{(2)} . a^{(1)} \left(\omega_{0,k}^{(1)} + \sum_{j=1}^{D} \omega_{j,k}^{(1)} v_{i,j} \right) \right). \tag{1.69}$$

This function can be represented in the form of a network diagram as shown in Figure 1.9. Then to evaluate the parameter ω we proceed to a forward and backward propagation of information through the network using typically the stochastic gradient descent, see the reference [17, 77].

The neural network model described above comprises just two stages of processing, where each step corresponds to a perceptron algorithm [16, 15]. Hence this neural network is referred as a multilayer perceptron (MLP). The first unit is called the input unit, the last is called the output unit, and between these two units we have what is called the hidden units. In the case where the activation functions of all the hidden units are just linear functions, the corresponding network is equivalent to a network without hidden units. This is due to the fact that the composition of linear operators gives a linear operator, so we just need to find one linear operator that will summarize all the hidden units. Moreover, if the number of hidden units is smaller than either the number of input or output units, the corresponding network simplifies the data by proceeding to a dimensionality reduction.

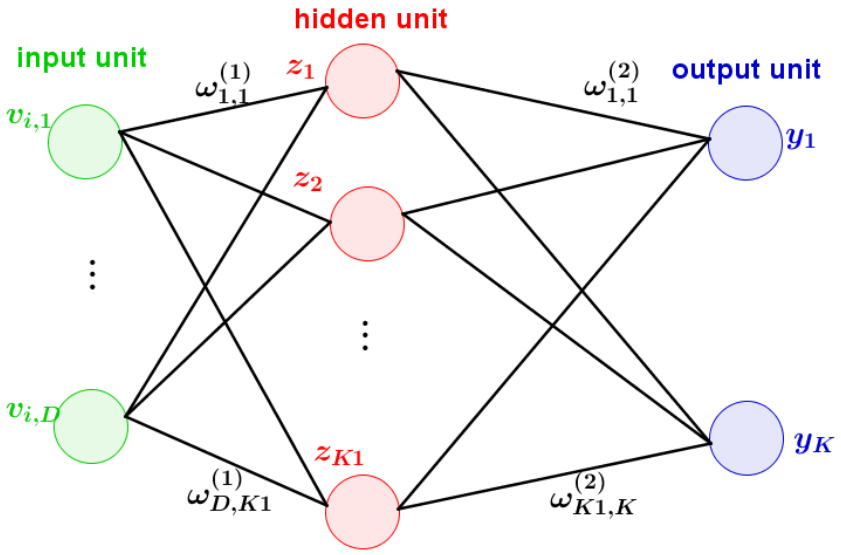


Figure 1.9: Illustration of the neural network with one input layer, one output layer and one hidden layer.

Biological motivation As explained in [72], the basic unit of the brain is a neuron, and our brain is composed of approximately 86 billion neurons. These neurons are connected thanks to the synapses, as illustrated in Figure 1.10. It has been inferred that each neuron receives an input signal from its dendrites and produces an output signal along one single axon. This axon can eventually be connected via synapses to the dendrites of different neurons. The basic idea is that the synaptic strengths play the role of the coefficient $\omega^{(l)}_{i,j}$ and they control the influence of the signal.

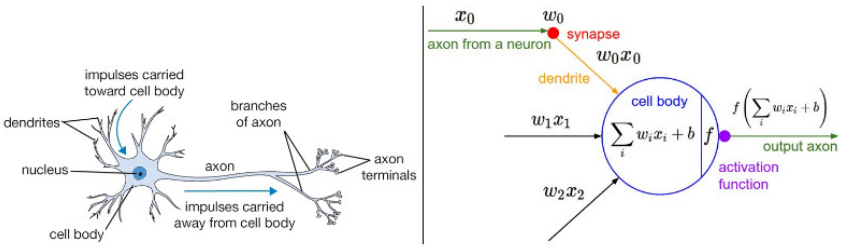


Figure 1.10: Illustration of the neural network from a biological point of view [72].

In the basic models, the signal is carried to the dendrites; once it is received, all the signals are sent to the cell body where they all get summed up. If the final sum is superior to a certain threshold, the neuron is activated, sending a spike along its axon. This can be interpreted as an activation function.

Convolutional Neural Network Neural networks are difficult to optimize (to learn) since we consider all the coordinates of v_i. That is why usually a sparse network that does not have all the possible connections between two layers is considered. One of this type of neural networks is underlying the paradigm of Convolutional Neural Networks (CNN). CNN are considered today as providing impressive improvement of state-of-the-are classification task in computer vision [74]. Right

now, one of the best results of classification are obtained with a CNN of 152 layers [65]. In these networks, the layer is organized into planes which can be seen as a feature map. Each layer takes as input only data from a small spatial subregion of the image as represented in Figure 1.11.

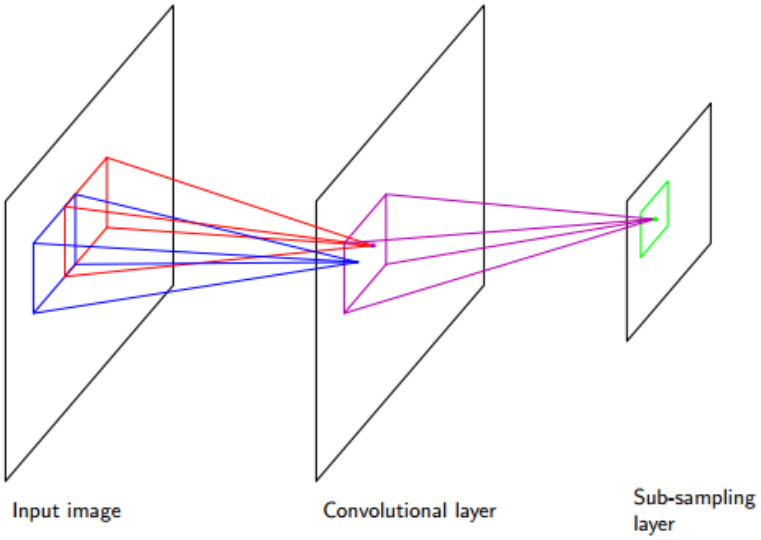


Figure 1.11: Illustration of a simple Convolutional Neural Network layer [16].

Let us use the notation of the "matconvnet library" [155] documentation in order to explain a bit more the CNN. An image is denoted as:

$$f : \begin{cases} E \longrightarrow F \\ (i,j) \longrightarrow f(i,j,:) \in \mathbb{R}^D \end{cases} \tag{1.70}$$

such that if we want to access to a particular feature coordinate k at a spatial position (i,j) we write it $f(i,j,k)$. A CNN can be seen as the composition of a given number of functions, each one of them corresponding to a layer:

$$y_{k1}(f) = g_N\left(\dots g_1\left(f, \omega^{(1)}\right), \omega^{(N)}\right) \tag{1.71}$$

Function g_l, $l = 1, \cdots, N$ takes as input the data ot the previous layer and a set of parameters $\omega^{(l)}$ which are learned from data in order to solve a target problem. There are different kinds of functions g_l as illustrated in Figure 1.12. Some g_l do not have any parameter.

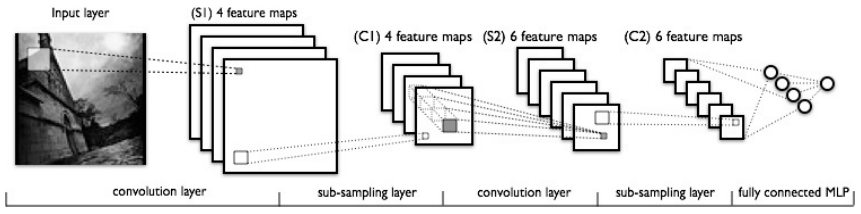


Figure 1.12: Illustration of a typical Convolutional Neural Network architecture [2].

The first function is typically the convolution. We note that here it is a three-dimensional convolution, in the sense that it operates on a spatial map taking into

consideration all the feature channels:

$$y(i',j',k')^{(1)} = g\left(f,\omega^{(1)}\right) = \sum_{i,j,k} \omega^{(1)}_{i,j,k,k'} f(i+i'-1, j+j'-1, k). \tag{1.72}$$

with $k' \in [1, N_1]$, and N_1 is the number of feature maps for layer 1. $y(i',j',k')^{(1)}$ represents the information of one feature maps k_1 of layer 1 in position (i',j'). $\omega^{(1)}$ represents a multi-dimensional filter. As we said, a nonlinearity is required to obtain useful features. The simplest nonlinear layer is a ReLU:

$$y(i',j',k')^{(2)} = g\left(y(i',j',k')^{(1)}\right) = \max(0, y(i',j',k')^{(1)}). \tag{1.73}$$

Next typical layers are so-called pooling layer. A pooling operator works on individual feature channels, replacing nearby features values by one feature thanks to a suitable operator. The most popular are the max-pooling and the mean-pooling. The max-pooling is defined by:

$$\begin{aligned} y(i,j,k)^{(3)} = g\left(y(:,:,k)^{(2)}\right) = \\ \max\{y(i',j',k)^{(2)} | i \leq i' \leq i+p \ \& \ j \leq j' \leq j+p\}, \end{aligned} \tag{1.74}$$

with p a scale parameter fixed by the user.

Chapter 2

Morphological Principal Component Analysis for Hyperspectral Image Analysis

Abstract

This chapter deals with the issue of reducing the spectral dimension of a hyperspectral image using principal component analysis (PCA). To perform this dimensionality reduction, we propose to add spatial information in order to improve the features that are extracted. Several approaches proposed to add spatial information are discussed in this chapter. They are based on mathematical morphology operators. These morphological operators are the area opening/closing, granulometries and grey-scale distance function. We name the proposed family of techniques the Morphological Principal Component Analysis (MorphPCA). Present approaches provide new feature spaces able to handle jointly the spatial and spectral information of hyperspectral images. They are computationally simple since the key element is the computation of an empirical covariance matrix which integrates simultaneously both spatial and spectral information. The performance of the different feature spaces is assessed for different tasks in order to prove their practical interest.

Résumé

Ce chapitre traite de la réduction de la dimension spectrale d'une image hyperspectrale à l'aide de l'analyse des composantes principales (PCA). Pour réaliser cette réduction de dimensionnalité, nous proposons d'ajouter des informations spatiales. Plusieurs approches ont été proposées pour ajouter de l'information spatiale dans ce chapitre. Elles sont basées sur des opérateurs de morphologie mathématique.

2.1 Introduction

Hyperspectral images allow us to reconstruct the spectral profiles of objects imaged by the acquisition of several tens or hundred of narrow spectral bands. Conventionally, in many applications hyperspectral images are reduced in the spectral dimension before any processing. Most of hyperspectral image reduction methods are linear and do not care about the multiple sources of nonlinearity present in this kind

of images [60]. Nonlinear reduction techniques are nowadays widely used on data reduction, and some of them have been used for hyperspectral images [9]. Nevertheless, most of these techniques present some disadvantages [154] in comparison to the canonical linear principal component analysis (PCA). That is the rationale behind our choice of PCA as starting point. In particular, one major drawback of those nonlinear techniques is that they are computationally too complex in comparison to PCA. Hence most of the time, they cannot be applied on real full resolution images. Another common disadvantage of both classical linear and nonlinear dimensionality reduction techniques is that they consider a hyperspectral image as a set of vectors. They are appropriate when the data do not present useful spatial information, and therefore they are not totally adapted to images.

As mentioned above, dimensionality reduction in hyperspectral images is usually considered as a preprocessing step for supervised pixel classification as well as for other hyperspectral image tasks such as unmixing, target detection, etc. Hence, our goal is to incorporate spatial information into the dimensionality reduction (DR).

The contribution of our approach can be summarized as follows. We propose to add spatial information on the estimation of the covariance matrix used for PCA computation. This is done by means of morphological image representations, which involve a nonlinear embedding of the original hyperspectral image into a morphological feature space.

Many previous works have considered how to introduce spatial information into hyperspectral dimensionality image reduction. We can divide these techniques into different fields. The first family of techniques is close to our paradigm since they are based on mathematical morphology

The rest of the chapter is organized as follows. Section 2 provides a remind on the mathematical morphology multi-scale representation tools used in our approach. Section 3 introduces in detail our approach named morphological principal component analysis (MorphPCA). In order to justify our framework, a summary of the classical theory underlying the standard PCA is provided as well as the notion of Pearson image correlation. Then, the four variants of MorphPCA are discussed, including an analysis of their corresponding covariance matrix meaning. The application of MorphPCA to hyperspectral dimensionality image reduction is considered in Section 4. That involves an assessment of the different variants according to different criteria. For some of the criteria, new techniques to evaluate the quality of dimensionality reduction techniques on image processing are introduced. Techniques arising from manifold learning are also considered in the comparaison. Finally, Section 5 closes the chapter with the conclusions.

The first data set was acquired over the city of Pavia (Italy), is a hyperspectral image of spatial size : 610×340 pixels, with 103 spectral bands. The second image, which represents the University of Houston, is a hyperspectral image of spatial size 349×1905 pixels and with 144 spectral bands [36]. The last one called Indian Pines is a hyperspectral image of spatial size 145×145 pixels, and with 224 spectral bands. We note that this chapter is an extended and improved version of our following contributions [49, 52].

2.2 Basics on morphological image representation

The goal of this section is to introduce a short background on morphological operators and transforms used in the sequel. Notation considered in the rest of this chapter is also stated.

2.2.1 Notation

Let E be a subset of the discrete space $\mathbb{Z}^2$, which represents the support space of a 2D image and $F \subseteq \mathbb{R}^D$ be a set of pixel values in dimension D. Hence, it is assumed in our case that the value of a pixel $x \in E$ is represented by a vector $v \in F$ of dimension D, where discrete space E has a size of $n_1 \times n_2$ pixels. This vector v represents the spectrum at position x. Additionally, we will write higher order tensors by calligraphic upper-case letters $(\mathcal{I}, \mathcal{S}, \ldots)$. The order of tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_J}$ is J. Moreover if $\mathcal{I} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, for all $i \in [1, n_3]$ $\mathcal{I}_{:,:,i}$ represents a matrix of size $n_1 \times n_2$ where the third component is equal to i. In our case we can also associate a tensor to the hyperspectral image $\mathcal{F} \in \mathbb{R}^{n_1 \times n_2 \times D}$.

2.2.2 Nonlinear scale-spaces and morphological decomposition

Mathematical morphology is a well known nonlinear image processing methodology based on the application of complete lattice theory to spatial structures. Let $f : E \to \mathbb{Z}$ be a grey-scale image. Area openings $\gamma^a_{s_l}(f)$ (resp. area closings $\varphi^a_{s_l}(f)$) are morphological filters that remove from the image f the bright (resp. dark) connected components having a surface area smaller than the parameter $s_l \in \mathbb{N}$ [160]:

$$\gamma^a_{s_l}(f) = \bigvee_i \{\gamma_{B_i}(f) | B_i \text{ is connected and } \mathrm{card}(B_i) = s_l\}, \tag{2.1}$$

$$\varphi^a_{s_l}(f) = \bigwedge_i \{\varphi_{B_i}(f) | B_i \text{ is connected and } \mathrm{card}(B_i) = s_l\}, \tag{2.2}$$

where $\gamma_B(f)$ and $\varphi_B(f)$ represent respectively the morphological flat opening and closing according to structuring element B [137]. We note that these connected filters can be implemented as binary filters on the stack decomposition of f into upper level sets. Figure 2.1 illustrates how area opening and area closing modify a simple image f. The image f in this toy example is composed of one black triangle of area equal to 30, 2 diamonds, one black and one white of area equal to 15. Finally the last connected components are 4 white circles and 5 black ones of area equal to 5. When an area opening is used (respectively closing) of threshold $s_l = 7$, just the white (respectively black) circles are removed.

Area opening and area closing are very relevant to simplify images, without deforming the contours of the objects remaining. In addition, area opening and closing can be used to produce a multi-scale decomposition of an image. The notion of morphological decomposition is related to the granulometry axiomatic [137]. Let us consider $\{\gamma^a_{s_l}\}$, $1 \leq l \leq S$ and $\{\varphi^a_{s_l}\}$, $1 \leq l \leq S$, two indexed families of area openings and closings respectively. Typically, the index l is associated to scale, or

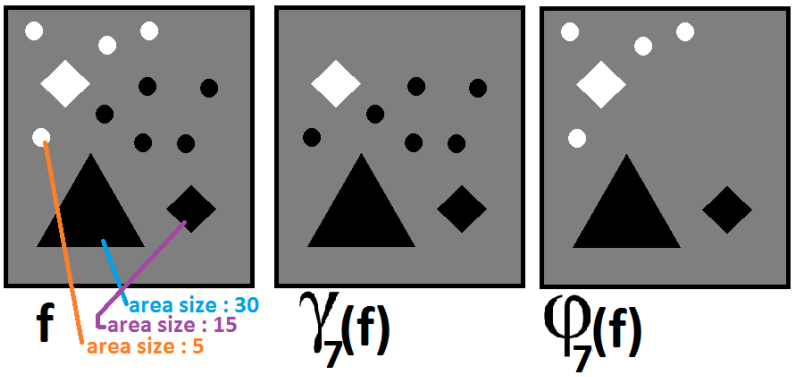


Figure 2.1: Illustration of an area opening $\gamma^a_{s_l}$ and an area closing $\varphi^a_{s_l}$ of image f, with $s_l = 7$ pixels. We can see that the connected components removed by the opening operator are the white circles since their area is 5, so below 7, and similarly for the black circles in the closed image.

more precisely to the surface area. Namely, we have on the one hand:

$$f = \sum_{l=1}^{S} (\gamma^a_{s_{l-1}}(f) - \gamma^a_{s_l}(f)) + \gamma^a_{s_S}(f); \tag{2.3}$$

$$f = \varphi^a_{s_S}(f) - \sum_{l=1}^{S} (\varphi^a_{s_l}(f) - \varphi^a_{s_{l-1}}(f)). \tag{2.4}$$

On the other hand, we can rewrite the decomposition [156]:

$$f = 1/2 \left((\gamma_{s_S}(f) + \varphi_{s_S}(f)) + \sum_{l=1}^{S} (\gamma^a_{s_{l-1}}(f) - \gamma^a_{s_l}(f)) - \sum_{l=1}^{S} (\varphi^a_{s_l}(f) - \varphi^a_{s_{l-1}}(f)) \right).$$

Therefore we have an additive decomposition of the initial image f into S scales, together with the average largest area opening and closing. We remark that the residue $(\gamma^a_{s_{l-1}}(f) - \gamma^a_{s_l}(f))$ represents bright details between levels s_l and s_{l-1}. Similarly, $(\varphi^a_{s_l}(f) - \varphi^a_{s_{l-1}}(f))$ stands for dark details between levels s_l and s_{l-1}. At this point, some issues must be taken into account. First, after decomposing an image into S scales, we have now to deal with an image representation of higher dimensionality. Second, the decomposition may not be optimal since it depends on the discretization of S scales, i.e., size of each scale. In order to illustrate that issue, we have represented in Figure 2.2(a) a channel of Pavia hyperspectral image and in Figure 2.2(b) its morphological decomposition by area openings that we have over-estimated. As it may be noticed from Figure 2.2(b), the choice of the scales is fundamental in order to avoid a redundant decomposition.

In order to deal with the problem of scale discretization, we propose to use the pattern spectrum that provides information about the image component size distribution. We can also notice another technique to find the optimal discretization [23].

2.2.3 Pattern Spectrum

The notion of pattern spectrum (PS) [99] corresponds to the probability density function (pdf) underlying a granulometric decomposition by morphological openings

and closings [103, 137]. The area-based PS of f at size s_l is given by

$$PS^a(f,l) = \left[\text{Mes}(\gamma^a_{s_l}(f) - \gamma^a_{s_{l+1}}(f))\right] / \text{Mes}(f), \tag{2.5}$$

$$PS^a(f,-l) = \left[\text{Mes}(\varphi^a_{s_{l+1}}(f) - \varphi^a_{s_l}(f))\right] / \text{Mes}(f), \tag{2.6}$$

where Mes represents here the integral of the grey-scale image. Two images having the same pattern spectrum have the same morphological distribution according to the choice of the family of openings/closings. Since our goal is to have a non-redundant multi-scale representation with the same morphological representation than the original image, then by sampling the PS and choosing the scales of the distribution which keep it as similar as possible to the image PS, we can expect to find the appropriate discretization of scales. However, one can see in Figure 2.3, the PS is not a smooth function, and consequently, sampling it with a limited number of scales, would not lead to a good result.

Based on the analogy between the PS and probability density function, we can compute its corresponding cumulative pattern spectrum (CPS) for both sides $l \geq 0$ and $l \leq 0$. Naturally, this function is smoother than the PS. In order to select the appropriate scales, the CPS for openings and closings are sampled, where the number of samples is fixed and is equal to S, under the constraint that the sampled function must be as similar as possible to the original function.

An example of such sampling is given in Figure 2.3, where the approximation of the CPS is depicted in red and the CPS of the original image in blue. It is well known in probability that two distributions that have the same cumulative distribution function have the same probability distribution function. Based on this property, we can expect that the discretization from the CPS approximates the original PS of the image and consequently, the selected scales represent properly the size distribution of the image.

2.2.4 Grey-scale distance function

Let X be the closed set associated to a binary image. The distance function corresponding to set X gives at each point $x \in X$ a positive number that depends on the position of x with respect to X and is given by [143]:

$$\text{dist}(X)(x) \;=\; \min\{d(x,y) : y \in X^c\}, \tag{2.7}$$

where $d(x,y)$ is the Euclidean distance between points x and y, and where X^c is the complement of set X. This well known transform is very useful in image processing [143].

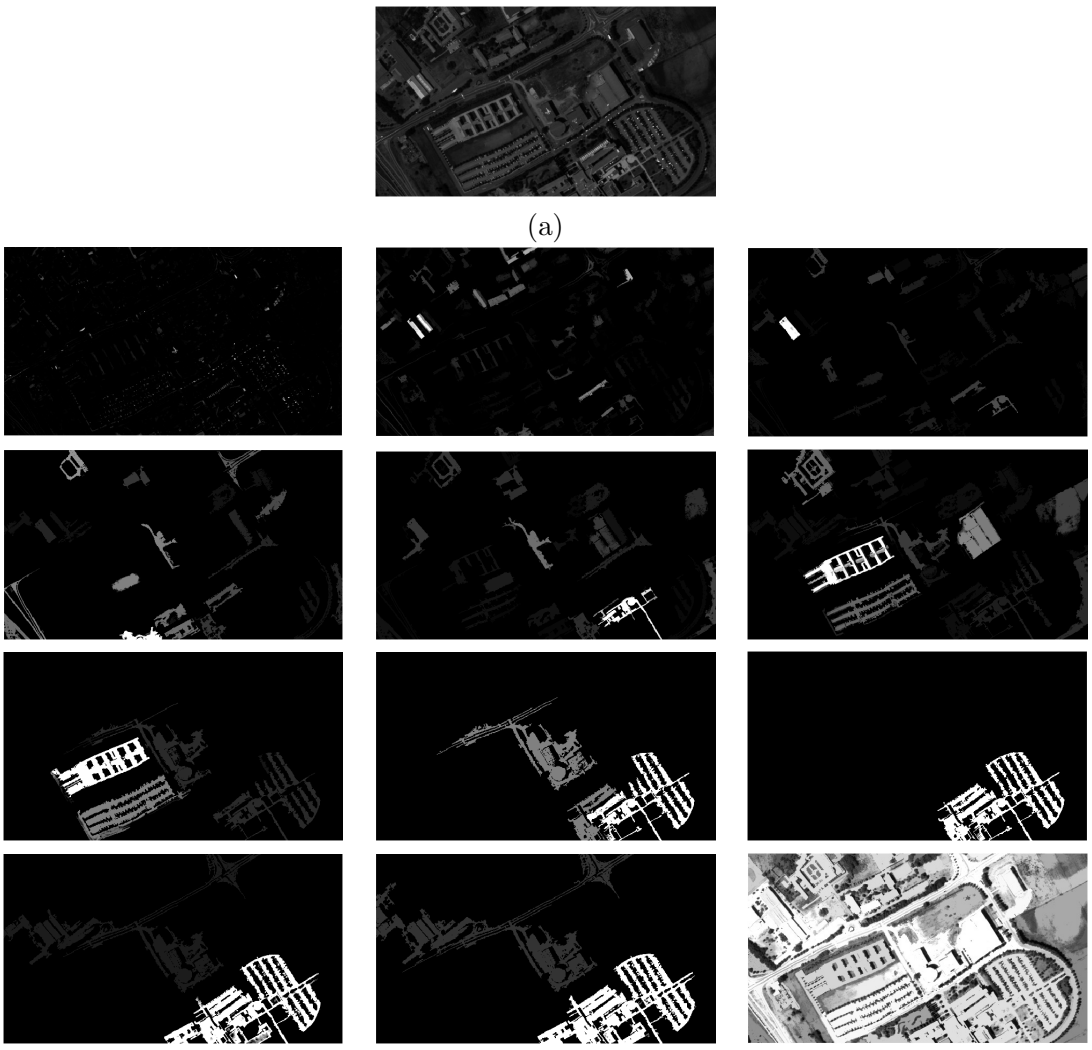
Distance function of binary images can be extended to grey-scale images f by considering its representation into upper level sets $\{X_h(f)\}_{a \leq h \leq b}$, where

$$X_h(f) = \{x \in E \;:\; f(x) \geq h\},$$

such that $a = \min\{f(x), x \in E\}$, and $b = \max\{f(x), x \in E\}$. Then, the so-called grey-scale distance transform of f is defined as [108]:

$$\text{dist}(f)(x) \;=\; (b-a)^{-1} \sum_{h=a}^{b} \text{dist}\left(X_h(f)\right)(x). \tag{2.8}$$

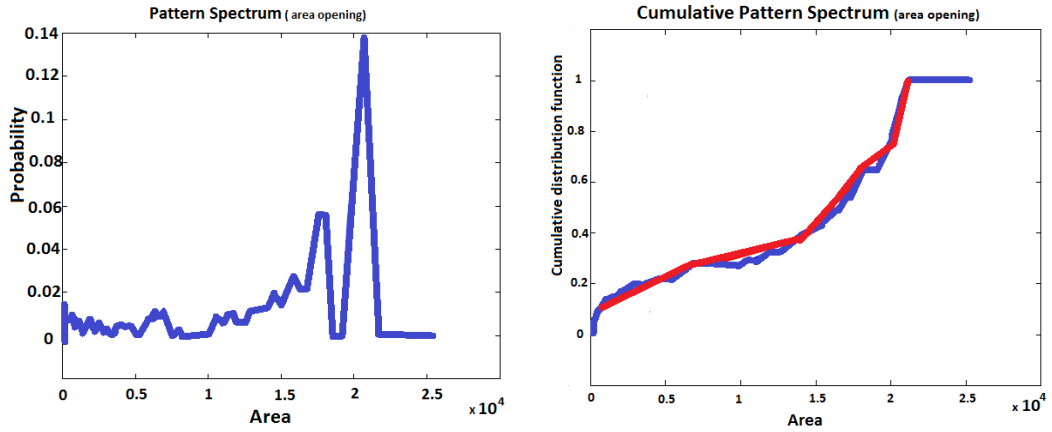
That is, the grey-scale distance transform of f is equal to the sum of the distance functions from its upper level sets.



(a)

(b)

Figure 2.2: (a) Channel number 50 of Pavia hyperspectral image and (b) its morphological decomposition by area openings $\gamma^a_{s_l}$, $s_l = \{0.5\ 10^2, 1\ 10^2, 5\ 10^2, 7\ 10^2, 1\ 10^3, 2\ 10^3, 5\ 10^3, 7\ 10^3, 1\ 10^4, 1.2\ 10^4, 1.5\ 10^4, 2.5\ 10^4\}$. Last image in (b) corresponds to $\gamma^a_{s_S}$, $s_S = 2.5\ 10^4$; the other images in (b) are $(\gamma^a_{s_{l-1}}(f) - \gamma^a_{s_l}(f))$. Note that the contrast of images has been enhanced to improve visualization.



(a) (b)

Figure 2.3: The pattern spectrum (PS) by area openings of a grey-scale image using 100 scales in (a). In (b), in blue, its corresponding cumulative pattern spectrum (CPS); in red, its approximation with $S = 8$ scales.

2.3 Morphological Principal Component Analysis

We introduce in this section the notion of Morphological Principal Component Analysis (MorphPCA) and its variants. Before reading this part we advice reader to read the section on the PCA the state of art chapter. On the next subsection we will focus on the covariance matrix needed to perform PCA.

2.3.1 Covariance matrix and Pearson correlation matrix

The covariance between two channels (or spectral bands) of an hyperspectral image $\mathcal{F}$ is computed as

$$\text{Covar}\left(\mathcal{F}_{:,:,k}, \mathcal{F}_{:,:,k'}\right) = \frac{1}{n}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\left[\mathcal{F}_{i,j,k} - \mathbb{E}(\mathcal{F}_{:,:,k})\right]\left[\mathcal{F}_{i,j,k'} - \mathbb{E}(\mathcal{F}_{:,:,k})\right], \tag{2.9}$$

where $\mathbb{E}(\mathcal{F}_{:,:,k})$ is the mean of the hyperspectral image channel k. The covariance is very meaningful, however this is not a similarity measure [56], in the sense of a metric, since it is not range limited. In order to fulfill this requirement, a solution consists in normalizing the covariance, which leads to the notion of Pearson correlation:

$$\text{Corr}\left(\mathcal{F}_{:,:,k}, \mathcal{F}_{:,:,k'}\right) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\left[\frac{\mathcal{F}_{i,j,k} - \mathbb{E}(\mathcal{F}_{:,:,k})}{\sigma_k}\right]\left[\frac{\mathcal{F}_{i,j,k'} - \mathbb{E}(\mathcal{F}_{:,:,k})}{\sigma'_k}\right], \tag{2.10}$$

where $\sigma_k = \left[\frac{1}{n}\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\left(\mathcal{F}_{i,j,k} - \mathbb{E}(\mathcal{F}_{:,:,k})\right)^2\right]^{1/2}$. The correlation coefficient varies between $+1$ and -1, such that $\text{Corr}\left(\mathcal{F}_{:,:,k}, \mathcal{F}_{:,:,k'}\right) = 1$ involves that $\mathcal{F}_{:,:,k}$ and $\mathcal{F}_{:,:,k'}$ perfectly coincide. It has been proved that the best fitting case corresponds to [70]:

$$\mathcal{F}_{i,j,k} = \text{Corr}\left(\mathcal{F}_{:,:,k}, \mathcal{F}_{:,:,k'}\right)\frac{\sigma_k}{\sigma'_k}(\mathcal{F}_{i,j,k'} - \mathbb{E}(\mathcal{F}_{:,:,k})) + \mathbb{E}(\mathcal{F}_{:,:,k}). \tag{2.11}$$

Therefore, from (2.11), we can see that the correlation is a linear coefficient between $\mathcal{F}_{i,j,k}$ and $\mathcal{F}_{i,j,k'}$. This means that Pearson correlation is a similarity criterion which depends on the intensities of the images and their linear relations.

2.3.2 MorphPCA and its variants

The fundamental idea of MorphPCA consists in replacing the covariance matrix V of PCA, which represents the statistical interaction of spectral bands, by a covariance matrix V_{Morpho} computed from a morphological representation of the bands. Therefore, mathematical morphology is fully integrated in the dimensionality reduction problem by standard SVD computation to solve

$$V_{\text{Morpho}} w_j = \lambda_j w_j.$$

The corresponding principal components w_j provide the projection space for the hyperspectral image $\mathcal{F}$. This principle is illustrated in the diagram of Figure 2.4.

We propose three variants of MorphPCA which are summarized in the flowchart of Figures 2.5, 2.6 and 2.8. An example of three different bands embedded in the space produced by these MorphPCA techniques is depicted in Figure 2.9.

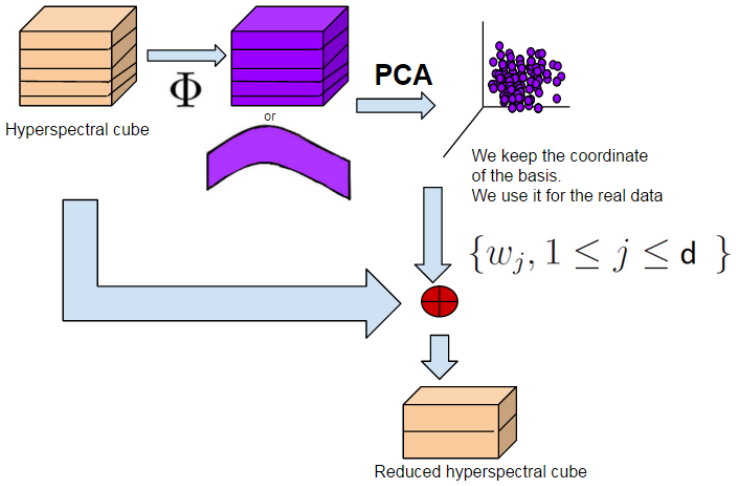


Figure 2.4: Global process of MorphPCA.

Scale-space Decomposition MorphPCA

In the first variant, we just use the area-based nonlinear scale-space discussed in the previous section. So the grey-scale image of each spectral band $\mathcal{F}_{:,:,k}$ is decomposed into residues of area openings and area closings according to the discretization into S scales for each operator, i.e., $r_l(\mathcal{F}_{:,:,k}) = \gamma^a_{s_{l-1}}(\mathcal{F}_{:,:,k}) - \gamma^a_{s_l}(\mathcal{F}_{:,:,k})$ and $r_{-l}(\mathcal{F}_{:,:,k}) = \varphi^a_{s_l}(\mathcal{F}_{:,:,k}) - \varphi^a_{s_{l-1}}(\mathcal{F}_{:,:,k})$, $1 \leq l \leq S$. Thus we have increased the dimensionality of the initial dataset from a tensor (n_1, n_2, D) to a tensor $(n_1, n_2, D, 2S+1)$. As discussed in [156], this tensor can be reduced using high order-SVD techniques. We propose here to simply compute a covariance matrix as the sum of the covariance matrices from the various scales. More precisely, we introduce $V_{\text{Morpho-1}} \in M_{D,D}(\mathbb{R})$ with :

$$V_{\text{Morpho-1}} = \sum_{l=1}^{S}(V(l)) + \sum_{l=1}^{S}(V(-l)) \tag{2.12}$$

where the covariance matrices at each scale l is obtained as

$$V(l)_{k,k'} = \text{Covar}\left(r_l(\mathcal{F}_{:,:,k}), r_l(\mathcal{F}_{:,:,k'})\right), \quad 1 \leq k, k' \leq D.$$

We note that it involves an assumption of independence of the various scales. We remark also that this technique is different from the classical approaches of differential profiles as [45] where the morphological decomposition is applied after computing the spectral PCA (i.e., morphology plays a role for spatial/spectral classification but not for spatial/spectral dimensionality reduction as in our case).

Pattern Spectrum MorphPCA

In the second variant, we can consider a very compact representation of the morphological information associated to the area-based nonlinear scale-space of each spectral band. It simply involves considering the area-based PS of each spectral band as the variable to be used to find statistical redundancy on the data. In other words, the corresponding covariance matrix $V_{\text{Morpho-2}} \in M_{D,D}(\mathbb{R})$ is defined as :

$$V_{\text{Morpho-2 } k,k'} = \text{Covar}\left(PS^a(\mathcal{F}_{:,:,k}, :), PS^a(\mathcal{F}_{:,:,k'}, :)\right), \tag{2.13}$$

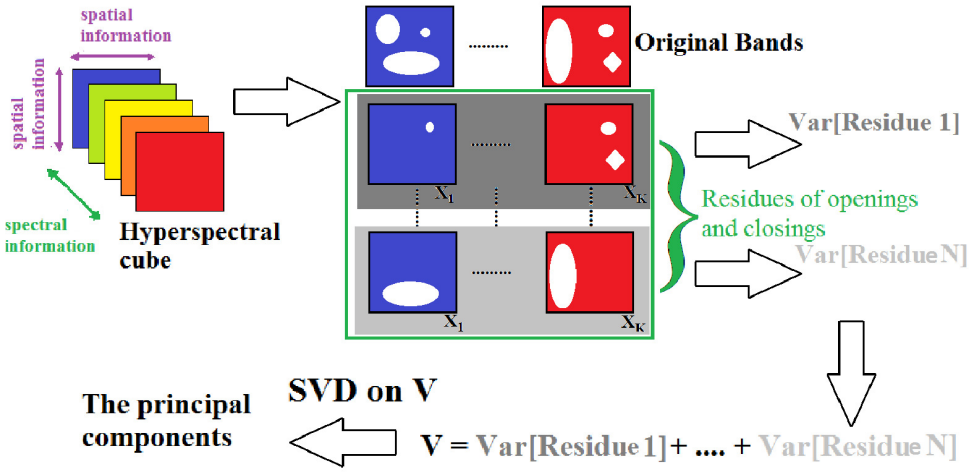


Figure 2.5: Process of scale-space decomposition MorphPCA.

with $1 \leq k, k' \leq D$ and where $PS^a(\mathcal{F}_{:,:,k}, l)$, $-S \leq l \leq S$, is the area-based pattern spectrum obtained by area-openings and area-closings. We note that the pattern spectrum can be seen as a kind of pdf of image structures. Consequently the MorphPCA associated to it explores the intrinsic dimensionality of sets of distributions instead of sets of vectors. For illustrating the information carried out by the PS, we have provided in Figure 2.9 the pattern spectra computed from three different bands of a hyperspectral image.

In order to better understand the interest of $V_{\text{Morpho-2}}$, we propose an analysis based on its Pearson correlation counterpart. Once the correlation of PS distribution is calculated, we have a linear coefficient between $PS^a(\mathcal{F}_{:,:,k}, l)$ and $PS^a(\mathcal{F}_{:,:,k'}, l)$. However since the PS is the result of nonlinear operations, the underlying extracted features are naturally nonlinear.

Let us consider the two binary images of Figure 2.7(a), which represent two objects having exactly the same size. If the correlations are calculated, we have:

$$\begin{aligned} \text{Corr}\,(\text{image1}, \text{image2}) &= 0 \\ \text{Corr}\,(PS^a(\text{image1}), PS^a(\text{image2})) &= 1. \end{aligned}$$

Hence, we can see that the morphological distribution being the same, the PS correlation is maximal. In a certain way, we observe that this transform builds size-invariants from the images and consequently it is robust to some groups of transforms and deformations. For instance, it is invariant to rotation and to translation.

Classical PCA on the spectral bands and the MorphPCA based on the PS can be compared by the corresponding correlation matrices from a hyperspectral image, such as the example plotted in Figure 2.10(a) and (b). From this visualization, we already observe that the bands are better discriminated between them.

Distance Function MorphPCA

Classical PCA for hyperspectral images is based on exploring covariances between spectral intensities. The previous MorphPCA involves changing the covariance into a morphological scale-space representation of the images. An alternative is founded

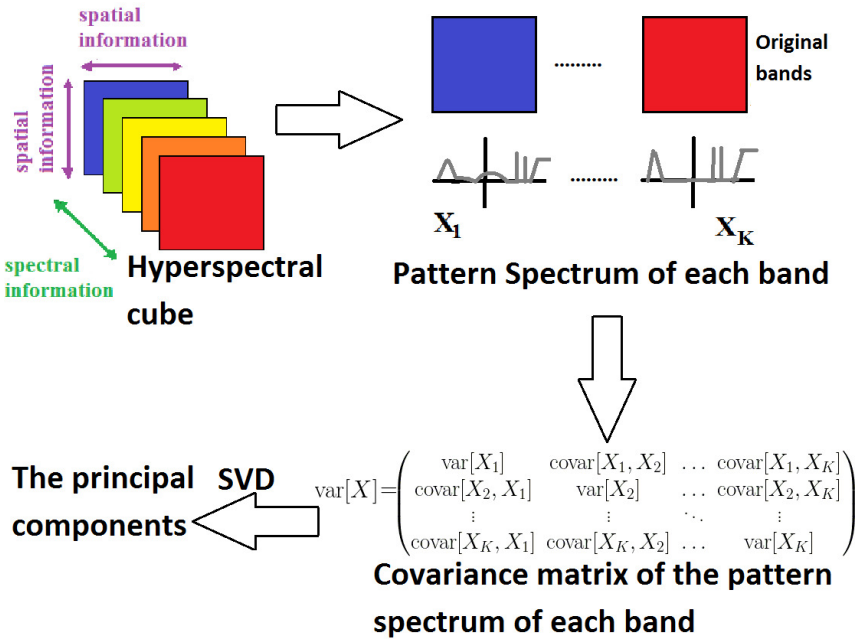
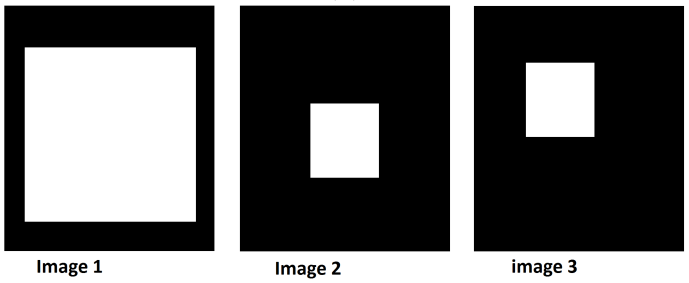


Figure 2.6: Process of pattern spectrum MorphPCA.

(a)



(b)

Figure 2.7: (a) Example of pair of binary images for pattern spectrum correlation discussion. (b) Example of triplet of binary images for distance function correlation discussion.

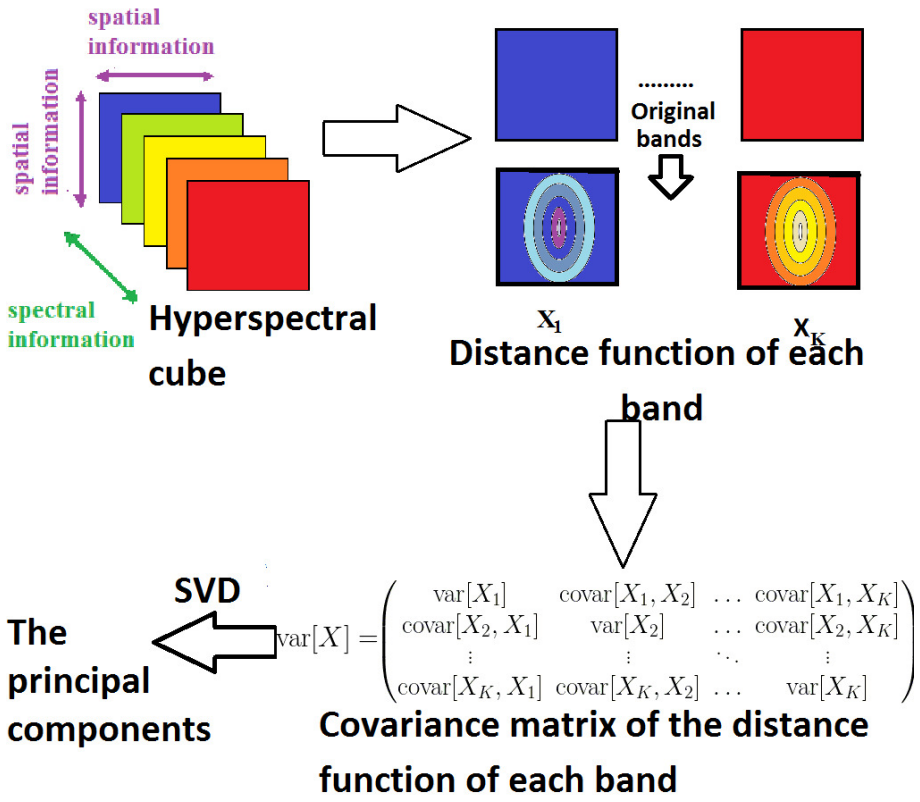


Figure 2.8: Process of distance function MorphPCA.

when transforming each spectral band from an intensity based map to a metric based map where at each pixel the value is associated to both the initial intensity and the spatial relationships between the image structures. This objective can be achieved using the Molchanov grey-scale distance function[108] for each spectral band $\mathrm{dist}(\mathcal{F}_{:,:,k})$. The new covariance matrix $V_{\text{Morpho-3}} \in M_{D,D}(\mathbb{R})$ is now defined as:

$$V_{\text{Morpho-3 } k,k'} = \mathrm{Covar}\left(\mathrm{dist}(\mathcal{F}_{:,:,k}), \mathrm{dist}(\mathcal{F}_{:,:,k'})\right), \tag{2.14}$$

with $1 \leq k, k' \leq D$. Figure 2.9 depicts the corresponding grey-scale distance function from three spectral band of a hyperspectral image. We note that this function carries out simultaneously both intensity and shape information from the image.

Let consider in detail the expression of the covariance of distance functions:

$$\begin{aligned} &\mathrm{Covar}\left(\mathrm{dist}(\mathcal{F}_{:,:,k}), \mathrm{dist}(\mathcal{F}_{:,:,k'})\right) = \\ &\mathrm{Covar}\left(\sum_{h=\min(\mathcal{F}_{:,:,k})}^{\max(\mathcal{F}_{:,:,k})} d(X_s(\mathcal{F}_{:,:,k})), \sum_{h'=\min(\mathcal{F}_{:,:,k'})}^{\max(\mathcal{F}_{:,:,k'})} d(X_{h'}(\mathcal{F}_{:,:,k'}))\right) = \\ &\sum_{h=\min(\mathcal{F}_{:,:,k})}^{\max(\mathcal{F}_{:,:,k})} \sum_{h'=\min(\mathcal{F}_{:,:,k'})}^{\max(\mathcal{F}_{:,:,k'})} \mathrm{Covar}\left(d(X_h(\mathcal{F}_{:,:,k})), d(X_{h'}(\mathcal{F}_{:,:,k'}))\right), \end{aligned}$$

where $X_h(\mathcal{F}_{:,:,k})$ denotes an upper level set at threshold h. The central term is the

covariance between two binary distance functions and can be developed as follows:

$$\begin{aligned}\text{Covar}\left(d(X_h(\mathcal{F}_{:,:,k})), d(X_{h'}(\mathcal{F}_{:,:,k'}))\right) = \\ \mathbb{E}\left(d(X_h(\mathcal{F}_{:,:,k})), d(X_{h'}(\mathcal{F}_{:,:,k'}))\right) - \mathbb{E}\left(d(X_h(\mathcal{F}_{:,:,k}))\right)\mathbb{E}\left(d(X_{h'}(\mathcal{F}_{:,:,k'}))\right) = \\ n^{-1} < d(X_h(\mathcal{F}_{:,:,k})), d(X_{h'}(\mathcal{F}_{:,:,k'})) >_{L^2} - \mathbb{E}\left(d(X_h(\mathcal{F}_{:,:,k}))\right)\mathbb{E}\left(d(X_{h'}(\mathcal{F}_{:,:,k'}))\right),\end{aligned}$$

where $< \cdot, \cdot >_{L^2}$ denotes the L^2 inner product. Using the classical relationship:

$$\|A - B\|_{L^2}^2 = \|A\|_{L^2}^2 + \|B\|_{L^2}^2 - 2 < A, B >_{L^2}, \ \forall A, B \in \mathbb{R}^n,$$

we finally obtain that:

$$\begin{aligned}\text{Covar}\left(d(X_h(\mathcal{F}_{:,:,k})), d(X_{h'}(\mathcal{F}_{:,:,k'}))\right) = (2n)^{-1}(\|d(X_h(\mathcal{F}_{:,:,k}))\|_{L^2}^2 + \|d(X_{h'}(\mathcal{F}_{:,:,k'}))\|_{L^2}^2) \\ -(2n)^{-1}\|d(X_h(\mathcal{F}_{:,:,k})) - d(X_{h'}(\mathcal{F}_{:,:,k'}))\|_{L^2}^2 - n^{-1}\|d(X_h(\mathcal{F}_{:,:,k}))\|_{L^2}^2\|d(X_{h'}(\mathcal{F}_{:,:,k'}))\|_{L^2}^2.\end{aligned}$$

From this latter expression, the term

$$\|d(X_h(\mathcal{F}_{:,:,k})) - d(X_{h'}(\mathcal{F}_{:,:,k'}))\|_{L^2}^2,$$

can be identified as the Baddeley distance [10] used in shape analysis. This distance is somehow equivalent to the most classical Hausdorff distance between the upper level sets h of spectral band k and h' of spectral band k'. Thus, the underlying similarity from this covariance compares the shape of the spectral channels, and extracts a richer description than Pearson correlation from the spectral channels themselves. We note that the use of Hausdorff distance between upper level sets of hyperspectral bands was previously used in [157].

Finally, to illustrate qualitatively the behavior of the distance function correlation, let us consider this time the three binary images depicted in Figure 2.7(b), where image 2 and image 3 represent the same object placed at a different location on the image. One has:

$$\begin{aligned}\text{Corr}\left(\text{image1}, \text{image2}\right) &= \text{Corr}\left(\text{image1}, \text{image3}\right) \\ \text{Corr}\left(\text{dist(image1)}, \text{dist(image2)}\right) &\neq \text{Corr}\left(\text{dist(image1)}, \text{dist(image3)}\right).\end{aligned}$$

That is, this similarity criterion related to the use of distance function is more discriminative to the relative position of the objects on the image than the classical Pearson Correlation.

From Figure 2.10(c), one can compare now the correlation matrix using the grey-scale distance function with the usual correlation matrix Figure 2.10(a). We note that this matrix provided also a better discrimination of bands cluster than the Pearson correlation matrix used in standard PCA.

Spatial/Spectral MorphPCA

As we have discussed, $V_{\text{Morpho-2}}$ represents a compact morphological representation of the image, however the spectral intensity information is also important for dimensionality reduction. To come with a last variant of MorphPCA, we build another covariance matrix $V_{\text{Morpho-4}}$ that represents the spectral and spatial information without increasing the dimensionality by the sum of two covariance matrices:

$$V_{\text{Morpho-4 }\beta} = (1 - \beta)V + \beta V_{\text{Morpho-2}}, \tag{2.15}$$

with $\beta \in [0, 1]$, and where obviously $V_{k,k'} = \text{Covar}\left(\mathcal{F}_{:,:,k}, \mathcal{F}_{:,:,k'}\right)$ and β stands for a regularization term that balances the spatial over the spectral information. This kind of linear combination of covariance matrices is similar to the one used in the combination of kernels, where kernels providing different information sources are combined to have a new kernel which integrates the various contributions [21].

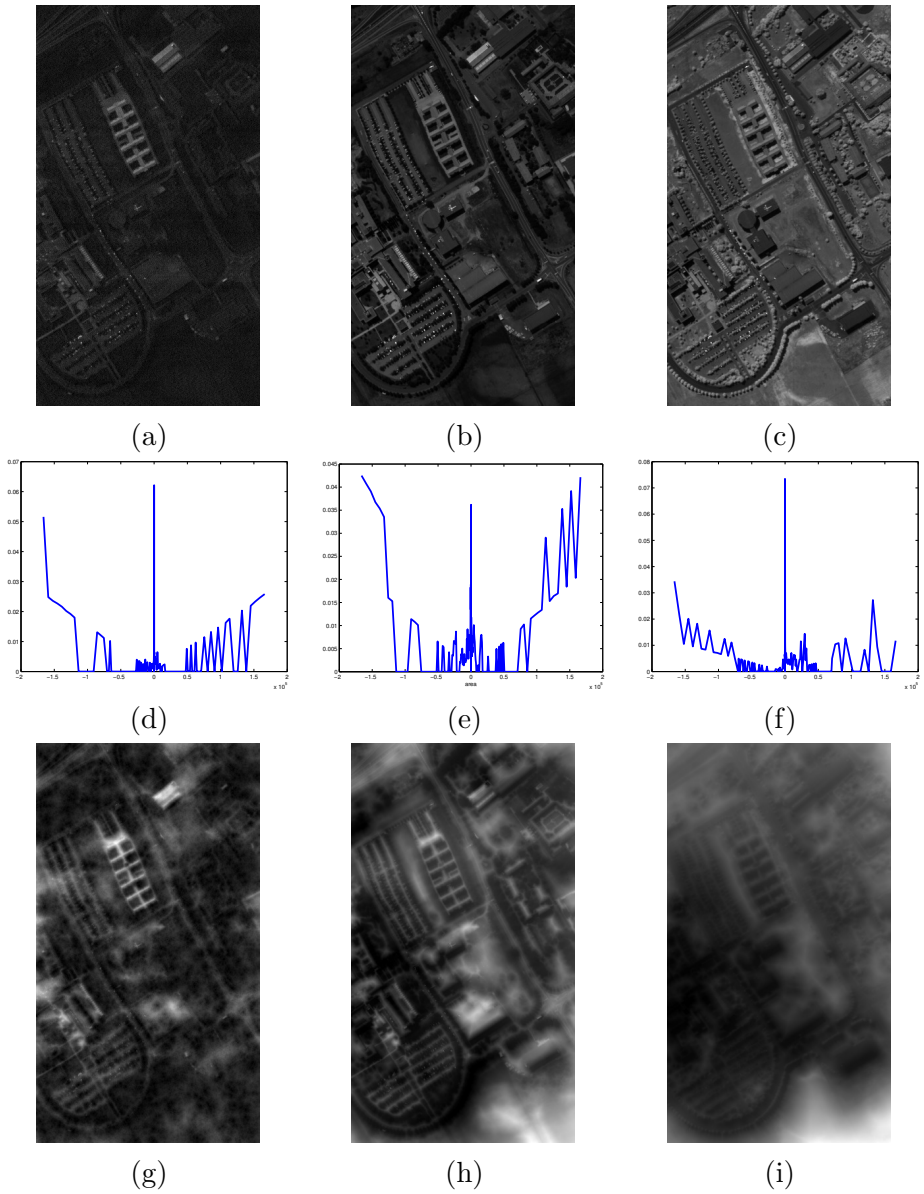
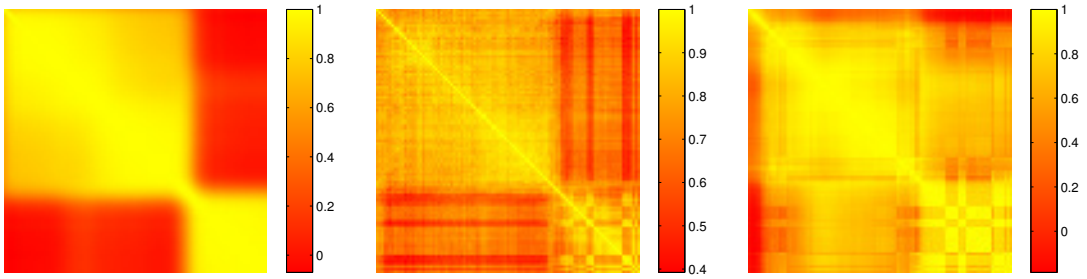


Figure 2.9: Top, three examples of spectral bands of Pavia image: (a) ♯1, (b) ♯50, (c) ♯100; middle, (d), (e), (f) PS of corresponding spectral bands; (g), (h), (i) Molchanov distance functions of corresponding spectral bands.



(a) (b) (c)

Figure 2.10: Visualization of the correlation matrix of (a) the spectral bands of Pavia hyperspectral image, (b) the PS of its spectral bands, (c) the distance function of its spectral bands.

2.4 MorphPCA applied to hyperspectral images

2.4.1 Criteria to evaluate PCA vs. MorphPCA

We can now use PCA and the four variants of MorphPCA to achieve dimensionality reduction (DR) of hyperspectral images. In order to evaluate the interest for such a purpose, it is necessary to establish quantitative criteria that should be assessed. These criteria will evaluate both locally and globally the effectiveness of the dimension reduction techniques.

- Local criteria.

  Criterion 1 (C1) The reconstructed hyperspectral image $\widetilde{\mathcal{F}}$ using the first d principal components should be a regularized version of $\mathcal{F}$ in order to be more spatially sparse.

  Criterion 2 (C2) The reconstructed hyperspectral image $\widetilde{\mathcal{F}}$ using the first d principal components should preserve local homogeneity and be coherent with the original hyperspectral image $\mathcal{F}$.

  Criterion 3 (C3) The manifold of variables (i.e., intrinsic geometry) from the reconstructed hyperspectral image $\widetilde{\mathcal{F}}$ should be as similar as possible to the manifold from original hyperspectral image $\mathcal{F}$.

- Global criteria.

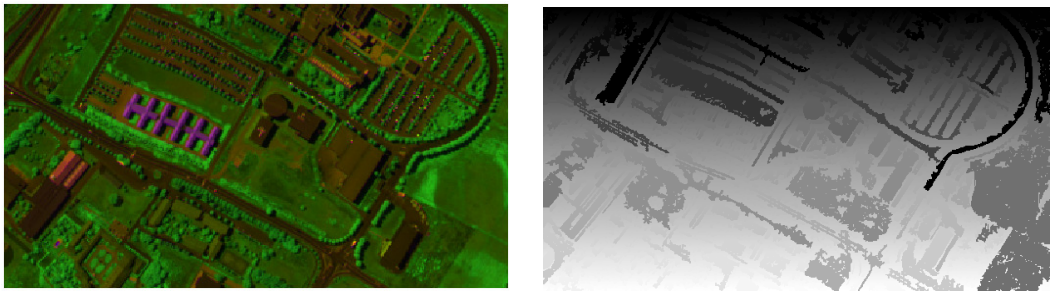
  Criterion 4 (C4) The number of bands d of the reduced hyperspectral image should be reduced as much as possible. It means that a spectrally sparse image is obtained.

  Criterion 5 (C5) The reconstructed hyperspectral image $\widetilde{\mathcal{F}}$ using the first d principal components should preserve the global similarity with the original hyperspectral image $\mathcal{F}$. Or in other words, it should be a good noise-free approximation.

  Criterion 6 (C6) Separability of spectral classes should be improved in the dimensionality reduced space. That involves in particular a better pixel classification.

These criteria are used to analyse the effectiveness of the DR methods studying locally and globally their ability to remove redundancy and to preserve the fully richness of the spectral and spatial information.

In order to assess C1, we compute the watershed transform [143] on each channel $\mathcal{F}_k$ of the hyperspectral image. Watershed transform is a morphological image segmentation approach which in a certain way can be seen as an unsupervised classification technique. The advantage of using the watershed is that it allows us to cluster the image according to the local homogeneity; thus, an image with less details will have less spatial classes than an image with many insignificant details. Then, the number of clusters $\|N_k\|$ of the watershed transform of $\mathcal{F}_k$ is considered as an estimation of the image complexity. To evaluate the complexity of the reconstructed hyperspectral image, the number of spatial classes is counted after having done a



(a) (b)

Figure 2.11: (a) A 3-variate image (first three eigenimages after PCA on Pavia hyperspectral image) and (b) its corresponding α-flat zone partition into 84931 spatial classes using the Euclidean distance.

watershed on each band. Finally, the mean of the number of spatial classes is taken, i.e.,

$$\text{Error}_{\text{sparse spatially}} = (D^{-1}) \sum_{k=1}^{D} \|N_k\|.$$

Assessment of C2, which involves image homogeneity, is based on a partition of the image into homogenous regions. Let us first remind the definition of a α-flat zones [106], used for such a purpose. Given a distance $d : \mathbb{R}^D \times \mathbb{R}^D \longrightarrow \mathbb{R}$, two pixels $(f(x), f(y)) \in (\mathbb{R}^D)^2$, from a vector-valued image f, belong to the same α-flat zone of f if and only if there is a path $(p_0, \ldots, p_n) \in E^n$ such as $p_0 = x$ and $p_n = y$ and $\forall i \in [1, n-1]$, $d\left(f(p_i), f(p_{i+1})\right) \leq \alpha$, with $\alpha \in \mathbb{R}^+$. Computing the α-flat zones for a given value of α produces therefore a spatial partition of the image into classes such that in each connected class the image values are linked by paths of local bounded variation. Working on the d eigenvectors, the image partition π_α associated to the α-flat zones quantize spatially and spectrally an hyperspectral image, see example given in Figure 2.11. The goal of simultaneous spatial and spectral quantization of a hyperspectral image has been studied in [53], where we have studied in detail the dependency on the distance. Moreover, we have shown that in high dimensional spaces quantization results are generally not good. For the case considered here, we propose to use the Euclidean distance on the reduced space by PCA or MorphPCA. The choice of α is done in order to guarantee a number C of α-flat zones similar for all the compared approaches. We can expect that, by fixing the number of zones in the partition, the difference between a partition and another one depends exclusively on the homogeneity of the image. Now, using the partition π_α, the spectral mean value of pixels from the original image $\mathcal{F}$ in each spatial zone is computed. This quantization produces a simplified hyperspectral image, denoted $\overline{\mathcal{F}}^{\pi_\alpha}$. Finally, we assess how far pixels of the original image from each α-flat zone are from their mean; which involves computing the following error

$$\text{Error}_{\text{Homg}} = \sum_{k=1}^{D} \sum_{i,j=1}^{n_1,n_2} |\mathcal{F}_{i,j,k} - \overline{\mathcal{F}}^{\pi_\alpha}_{i,j,k}|^2.$$

This criterion can consequently be seen as a way to see the trustworthiness of the

DR technique, since it measures if the homogeneous partition of the reduced hyperspectral image corresponds to the homogeneous zone of the original image.

C3 has been evaluated by means of two manifold learning criteria called the K-intrusion and K-extrusion [79]. They are based on other criteria called continuity and trustworthiness [159]. These criteria reveal DR behavior in terms of its ability to preserve the data manifold structure. We have first sampled randomly 10 thousands spectra from our hyperspectral images, where each spectrum is a vector of dimension D. Then we have modelled the manifold by a graph where each node is a vector and each edge is the pairwise distance. We used the Euclidean distance as the pairwise distance. For the rest of the paragraph we note by x_i a point from the original manifold, ν_i^K its neighbourhood of size K, $\tilde{x_i}$ the same point from the manifold after a DR and $\tilde{\nu}_i^K$ its corresponding neighbourhood of size K. A neighbourhood of size K at point x_i is composed of the K closest points to x_i according to used metric. More precisely, the goal of K-extrusion is to measure how the points that were in the K-neighbourhood of x_i are not preserved in the K-neighbourhood of $\tilde{x_i}$ after DR. The K-intrusion evaluates if points on the K-neighbourhood of $\tilde{x_i}$ on the DR manifold were in the K-neighbourhood of x_i, i.e.,

$$M_{\text{intrusion}}(K) = 1 - \frac{2}{G(K)} \times \sum_{i=1}^{n} \sum_{j \in \tilde{\nu}_i^K \setminus \nu_i^K} r(i,j) - K, \tag{2.16}$$

$$M_{\text{extrusion}}(K) = 1 - \frac{2}{G(K)} \times \sum_{i=1}^{n} \sum_{j \in \nu_i^K \setminus \tilde{\nu}_i^K} r(\tilde{i},j) - K, \tag{2.17}$$

where $r(i,j)$ is the rank of the data x_j in the ordering according to the distance from x_i, and respectively $r(\tilde{i},j)$ the rank of $\tilde{x_j}$ in the ordering according to the distance from $\tilde{x_i}$, and the term G_K scales the measure to be between zero and one, i.e.,

$$G(K) = \begin{cases} NK(2N-3K-1) \text{ if } K < N \setminus 2 \\ N(N-K)(N-K-1) \text{ if } K \geq N \setminus 2. \end{cases} \tag{2.18}$$

For a better understanding of these formulae, see [158]. An important point is the dependence of these parameters on the size of the neighbourhood. From (2.16) and (2.17), the following parameters are computed[158]:

$$Q(K) = \frac{M_{\text{extrusion}}(K) + M_{\text{intrusion}}(K)}{2}, \tag{2.19}$$

$$B(K) = M_{\text{intrusion}}(K) - M_{\text{extrusion}}(K). \tag{2.20}$$

The interest of $Q(K)$ is that it estimates in average the quality of a DR technique, whereas $B(K)$ reveals its behavior as being more intrusive or extrusive.

In order to assess C4, as classically done, the fraction of explained covariance is fixed. Then, the number of principal components needed is counted. The rationale is based on the fact that a good DR technique should reduce the number of dimensions and extract a limited number of features that would explain most of the image. However since this criterion is linked to a sparsity criterion, we would like to add a distortion criterion, C5.

The evaluation of C5 is founded on computing a pattern spectrum of both the original hyperspectral image and the DR image. An important point is that the

pattern spectrum will be computed by openings on the hyperspectral image viewed as a 3D image. By doing such assumption, the 3D openings are decomposing in a simultaneous way the spatial/spectral object of the image and the corresponding curves of the PS will represent the distribution of both the spatial and the spectral objects. Two hyperspectral images are similar if they have the same spectral/spatial size distribution. As discussed in Section 2, we prefer to use the cumulative PS in order to obtain a smoother curve. Normally we cannot deal with both spatial/spectral distortions with the reconstruction error of the two images. However we will also assess the SNR of the reconstruction error as an additional parameter.

Finally, C6 is related to supervised pixel classification of the hyperspectral image. We have considered the least square SVM algorithm [20] as a learning technique, with a linear kernel or rbf kernel, where the rbf kernel is initialized for each DR technique using cross validation. For each supervised classification run, we used for the AVIRIS Indian Pine Image 5% of the available data as a training set and the remaining 95% to validate. For the ROSIS Pavia University image we use a subset of 50 spectra (about 1% of the available data) per class as a training set and the remaining spectra to validate.

2.4.2 Evaluation of algorithms

The studied DR techniques presented are listed and compared upon three mathematical and computing properties in Table 2.1. These properties were also considered in the excellent comparative review [154]. For comparison, we have also included in the table the Kernel-PCA (KPCA), which is a powerful generalization of PCA allowing integrating morphological and spatial features into DR.

The first one is the number of free parameters to be chosen. The interest of having these free parameters is that it provides more flexibility to the techniques, whereas the related inconvenient is the difficulty for properly tuning the right parameter. We notice that KPCA provides good flexibility thanks to the choice of any possible kernel which fits the data geometry. The most simple algorithms are the PCA, and the distance function MorphPCA. Then, we have the scale-space decomposition MorphPCA, and finally the pattern spectrum MorphPCA. The second issue analysed is the computational complexity, and the third one is the memory requirements. From a computational viewpoint, the most demanding step in the PCA is the SVD, which can be done in $O(D^3)$. PCA is the technique with the smallest computational need. On the contrary, the computational requirement of KPCA is $O(n^3)$; since $n \gg D$, this kind of algorithm seems infeasible in standard hyperspectral images. That is the reason why most of hyperspectral KPCA techniques use tricks to be able to deal with the high number of spectra [104, 107, 63]. All these techniques lead to a spatial distortion, which is not avoidable by the need of a sampling procedure aiming at reducing the number of spectra. Between the complexity of PCA and KPCA, we have our proposed MorphPCA algorithms. Regarding MorphPCA, the computationally demanding step is the computation of the morphological representation used in the corresponding covariance matrix. The complexity estimation has been carried out each time in the worse case, however efficient morphological algorithms can improve this part. Distance function MorphPCA is last demanding, then the scale-space decomposition MorphPCA and finally the pattern spectrum MorphPCA. Regarding memory needs, for the PCA, the pattern spectrum MorphPCA, and the

distance function MorphPCA, the steps requiring more memory is the storage of the covariance matrix, just of $O(D^2)$. The Spatial/Spectral MorphPCA needs to store 2 covariance matrices then its memory need is $O(2D^2)$; similarly the scale-space MorphPCA needs to store $2S+1$ covariance matrices, then its required memory is $O((2S+1)D^2)$. Note that KPCA uses a Gram matrix of size $(n \times n)$.

Technique	**Parameter (1)**	**Computational (2)**	**Memory (3)**
PCA	Prop	$O(D^3)$	$O(D^2)$
MorphPCA Morpho-1	Prop, S	$O(DnS_S(2S+1))$	$O(D^2(2S+1))$
MorphPCA Morpho-2	Prop, S	$O(DnS_S(2S+1))$	$O(D^2)$
MorphPCA Morpho- 3	Prop	$O(Dn(b-a))$	$O(D^2)$
MorphPCA Morpho-4 $_\beta$	Prop, β	$O(DnS_S(2S+1))$	$O(2D^2)$
KPCA	Prop, K	$O(n^3)$	$O(n^2)$

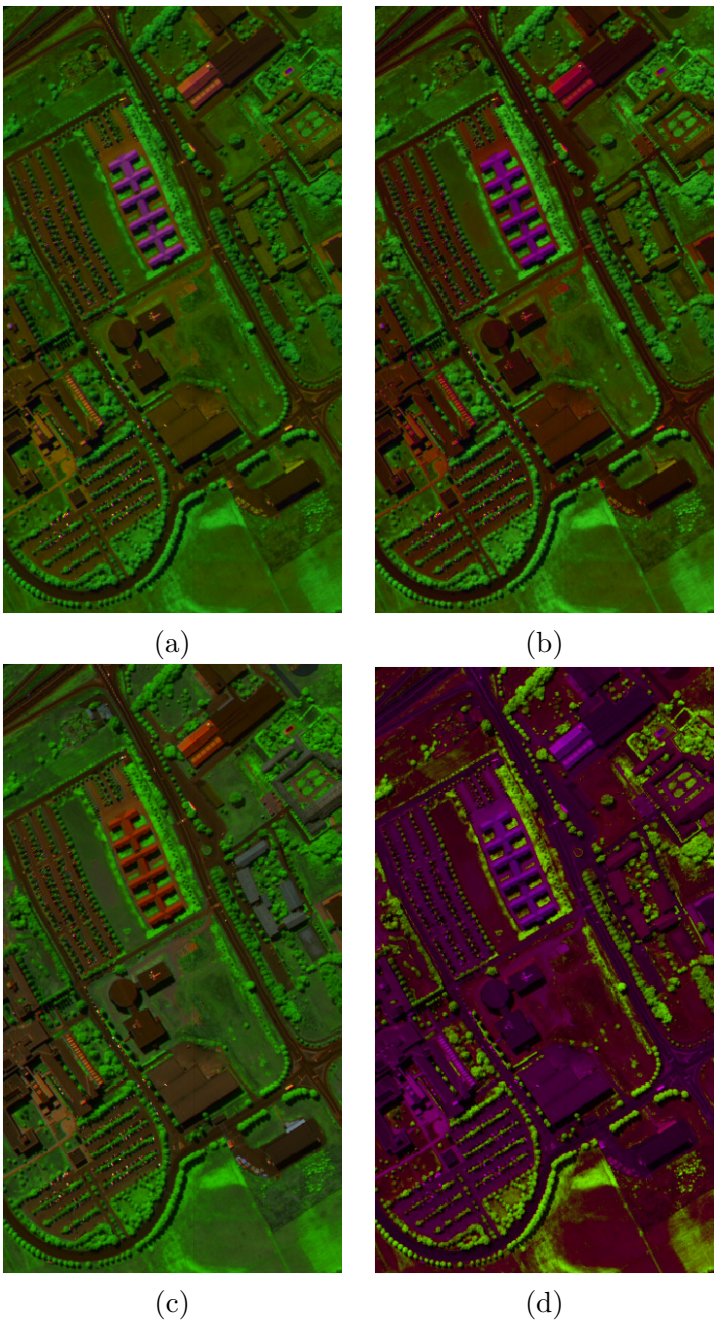
Table 2.1: Comparison of the properties of dimensionality reduction algorithms for hyperspectral images.

2.4.3 Evaluation on hyperspectral images

The assessment of the performance of PCA and MorphPCA has been carried out on three hyperspectral images. The first image was acquired over the city of Pavia (Italy) and it represents the university campus. The dimensions of the image are 610×340 pixels, with $D = 103$ spectral bands and its geometrical resolution is of 1.3 m. We also used a second hyperspectral image which represents the University of Houston campus and the neighbouring urban area at the spatial resolution of 2.5 m and which dimensions are 349×1905 pixels and $D = 144$ spectral bands [36]. The third image, acquired over the region of the Indian Pines test site in Northwestern Indiana, is composed for two-thirds of agriculture, and one-third of forest. The dimensions of this image are 145×145 pixels, $D = 224$ spectral bands and its geometrical resolution is of 3.7 m.

We have applied classical PCA and the different variants of MorphPCA to Pavia hyperspectral image. Figure 2.12 shows the first three eigenimages, visualized as a RGB false color. We note that the pattern spectrum MorphPCA requires $d = 5$ to represent 92% of the variance whereas the other approaches only impose $d = 3$. An interesting aspect observed on the projection of the 103 spectral channels of Pavia hyperspectral image into the first two eigenvectors is how PCA and the scale-space decomposition MorphPCA cluster the bands linearly, see Figure 2.13(a) and 2.13(b). Bands close in the projection are also near in the spectral domain, whereas the pattern spectrum MorphPCA, 2.13(c), and distance function MorphPCA, 2.13(d), tend to cluster spectral bands which are not necessary spectrally contiguous. This can be explained thanks to Figure 2.10, where the MorphPCA correlation matrices are different from the classical PCA one.

It can be noticed that in classical manifold learning techniques, the goal is to decrease the dimension of the data while keeping some properties on the data manifold. We work here on the manifold of the channels. This manifold is easier to use, but finding the good β in $V_{\text{Morpho-4 }\beta}$ that would maintain some properties of



(a) (b)

(c) (d)

Figure 2.12: RGB false color visualization of first three eigenimages from Pavia hyperspectral image: (a) classical PCA on spectral bands, (b) scale-decomposition MorphPCA, (c) pattern spectrum MorphPCA, (d) distance function MorphPCA.

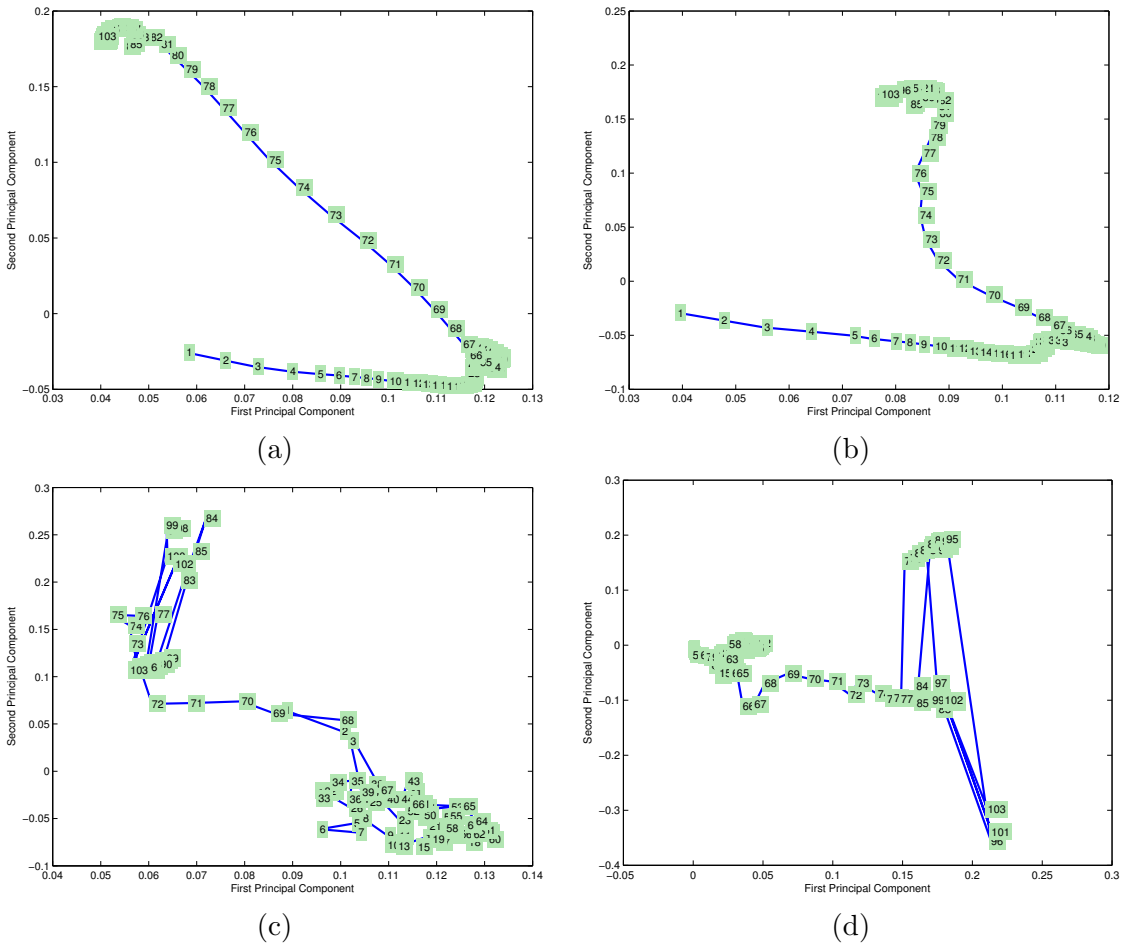


Figure 2.13: Hyperspectral band projection into the first two eigenvectors (i.e., image manifold) from Pavia hyperspectral image: (a) classical PCA on spectral bands, (b) scale-decomposition MorphPCA, (c) pattern spectrum MorphPCA, (d) distance function MorphPCA.

the manifold is not always easy, since we had to deal with a double optimization problem, i.e., β and d.

From a quantitative viewpoint, one can see in Table 3.1 that globally MorphPCA produces a more homogenous regularization of the image than classical PCA, especially the distance function MorphPCA and Spatial/Spectral MorphPCA with an appropriate $\beta = 0.2$, which gives the lowest values of $\text{Error}_{\text{Homg}}$. We noted that $\text{Error}_{\text{sparse spatially}}$ follows a different ranking. A good method is the one with a good trade-off between both criteria, since one wants a DR to be trustworthy, which is evaluated thanks to $\text{Error}_{\text{Homg}}$. But if the signal is too noisy, one may prefer a sparser representation. According to these criteria, the distance function MorphPCA and the pattern spectrum MorphPCA seem to have the best result. We can also note that if we use manifold learning parameters for criterion C3, see Figure 2.14, the pattern spectrum MorphPCA has the best results.

	V	$V_{\textbf{Morpho-1}}$	$V_{\textbf{Morpho-2}}$	$V_{\textbf{Morpho-3}}$
$\text{Error}_{\textbf{Homg}}$	100	100	95.9	79.3
$\text{Error}_{\textbf{sparse spatially}}$	99.8	99.7	100	88.3
	$V_{\textbf{Morpho-4}\ \beta}$ $\beta = 0.8$	$V_{\textbf{Morpho-4}\ \beta}$ $\beta = 0.2$	$V_{\textbf{Morpho-4}\ \beta}$ $\beta = 0.5$	
$\text{Error}_{\textbf{Homg}}$	93.2	83.9	88.3	
$\text{Error}_{\textbf{sparse spatially}}$	93.3	96.7	98,6	

(a)

	V	$V_{\textbf{Morpho-1}}$	$V_{\textbf{Morpho-2}}$	$V_{\textbf{Morpho-3}}$
$\text{Error}_{\textbf{Homg}}$	100	90.4	35.3	38.3
$\text{Error}_{\textbf{sparse spatially}}$	97.7	97.6	100	89

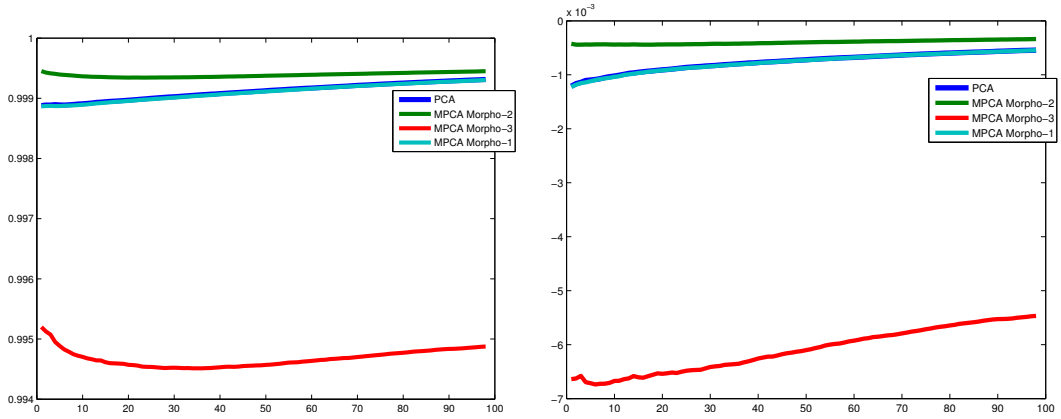
(b)

	V	$V_{\textbf{Morpho-1}}$	$V_{\textbf{Morpho-2}}$	$V_{\textbf{Morpho-3}}$
$\text{Error}_{\textbf{Homg}}$	98.1	100	96.5	97.8
$\text{Error}_{\textbf{sparse spatially}}$	91	100	91.2	82.7

(c)

Table 2.2: Comparison of PCA and MorphPCA analysis using criteria C1 and C2: (a) for Pavia hyperspectral image, (b) for Houston hyperspectral image, (d) for Indian Pines hyperspectral image. The values have been normalized to the worst case, which gives 100.

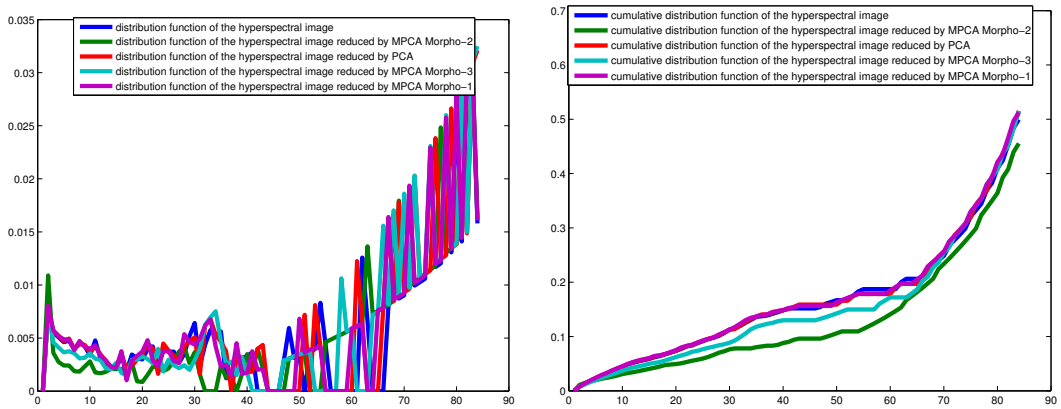
With respect to criterion C5, we have computed the 3D pattern spectrum distribution of Pavia hyperspectral image and of the different reduced images into d components, see Figure 2.15. From this result, we can see that both PCA and scale-space decomposition MorphPCA follow very well the hyperspectral image, since their spatial and spectral cumulative distributions are similar. However if one would like to denoise the hyperspectral image thanks to a DR technique, these results are not always positive. If we compare the spatial and spectral cumulative distribution of the distance function MorphPCA and the one of the hyperspectral image, we notice that for small 3D size (i.e., small spatial/spectral variations) that can be considered as noise, there are differences between these two distributions. But when the size



(a) (b)

Figure 2.14: Intrusion/Extrusion parameters for PCA and the different variants of MorphPCA from Pavia hyperspectral image: (a) $Q(K)$, (b) $B(K)$.

increases, the distribution of the the distance function MorphPCA tends to the hyperspectral image one. So it seems that the distance function MorphPCA simplifies the spectral/spatial noise, considered as the small 3D objects, but keeps the objects of interest.



(a) (b)

Figure 2.15: (a) 3D pattern spectrum distribution of Pavia hyperspectral image and of the different reduced images into d components. (b) Corresponding 3D cumulative pattern spectrum distributions.

Finally, Table 3.2 summarizes the results of supervised classification of respectively Pavia and Indian Pine hyperspectral images. We note that results for Pavia image are quite similar in all the cases even if MorphPCA seems to be better than PCA. Therefore, we have chosen to focus on the Indian Pine image, which is more challenging for supervised classification benchmark, see also Figure 2.16 and 2.17. We note that MorphPCA improves the results, especially the scale-space decomposition MorphPCA. To evaluate the classification results, first we fixed the dimension d of the reduce image. We have chosen $d = 5$. Then, we used the least square SVM, which is a multi-class classification technique, contrary to classical two-class SVM.

We also used two simple kernels: the linear one, which is the simplest one, and the rbf one, which is appropriate for hyperspectral images since we can assume that these data follow Gaussian distribution. Finally, we study the influence of dimension d and of the size of the training set on the different DR techniques over the classification results. For this purpose we have depicted in Figure 2.18 and Figure 2.19 the evolution of the kappa statistics. From the latter plot we can see that the PCA and the pattern spectrum PCA have the worst results. By combining the spectral and the spatial information, a better classification can be achieved. This is the case of the distance function MorphPCA, the scale-space decomposition MorphPCA, and the Spatial-Spectral MorphPCA for $\beta = 0.2$.

	Overall Accuracy with linear kernel	**Overall Accuracy with rbf kernel**	**Kappa statistic with rbf kernel**
V	51.51 ± 0.9	84.9 ± 3.1	$0.84 \pm 1 \times 10^{-4}$
$V_{\textbf{Morpho-1}}$	59.6 ± 2.2	85.8 ± 2.6	$0.84 \pm 1 \times 10^{-4}$
$V_{\textbf{Morpho-2}}$	56.99 ± 1.1	85.2 ± 2.1	$0.84 \pm 1 \times 10^{-4}$
$V_{\textbf{Morpho-3}}$	59.9 ± 2.5	86.0 ± 1.9	$0.84 \pm 1 \times 10^{-4}$
$V_{\textbf{Morpho-4}\ \beta}$, $\beta = 0.2$	61.0 ± 1.73	85.2 ± 1.1	$0.83 \pm 1 \times 10^{-4}$
$V_{\textbf{Morpho-4}\ \beta}$, $\beta = 0.5$	59.9 ± 1.5	84.6 ± 1.0	$0.83 \pm 1 \times 10^{-4}$
$V_{\textbf{Morpho-4}\ \beta}$, $\beta = 0.8$	57.87 ± 3	84.7 ± 2.5	$0.83 \pm 2 \times 10^{-4}$

(a) Pavia image

	Overall Accuracy with linear kernel	**Overall Accuracy with rbf kernel**	**Kappa statistic with rbf kernel**
V	43.9 ± 3.6	75.2 ± 3.7	$0.73 \pm 4.3 \times 10^{-4}$
$V_{\textbf{Morpho-1}}$	50.5 ± 3.8	79.6 ± 3.7	$0.78 \pm 4 \times 10^{-4}$
$V_{\textbf{Morpho-2}}$	41.5 ± 3.8	66.6 ± 4.6	$0.63 \pm 4.5 \times 10^{-4}$
$V_{\textbf{Morpho-3}}$	51.3 ± 3.2	79.1 ± 3.2	$0.77 \pm 3.7 \times 10^{-4}$
$V_{\textbf{Morpho-4}\ \beta}$, $\beta = 0.2$	43.5 ± 3.3	75.1 ± 2.3	$0.72 \pm 2.6 \times 10^{-4}$
$V_{\textbf{Morpho-4}\ \beta}$, $\beta = 0.5$	43.1 ± 2.9	71.2 ± 2.6	$0.68 \pm 3 \times 10^{-4}$
$V_{\textbf{Morpho-4}\ \beta}$, $\beta = 0.8$	43.0 ± 2.2	69.7 ± 3.3	$0.67 \pm 3.9 \times 10^{-4}$

(b) Indian Pine image

Table 2.3: Comparison of hyperspectral supervised classification on PCA and MorphPCA spaces using least square SVM algorithm and different kernels: (a) Pavia hyperspectral image, (b) Indian Pines hyperspectral image.

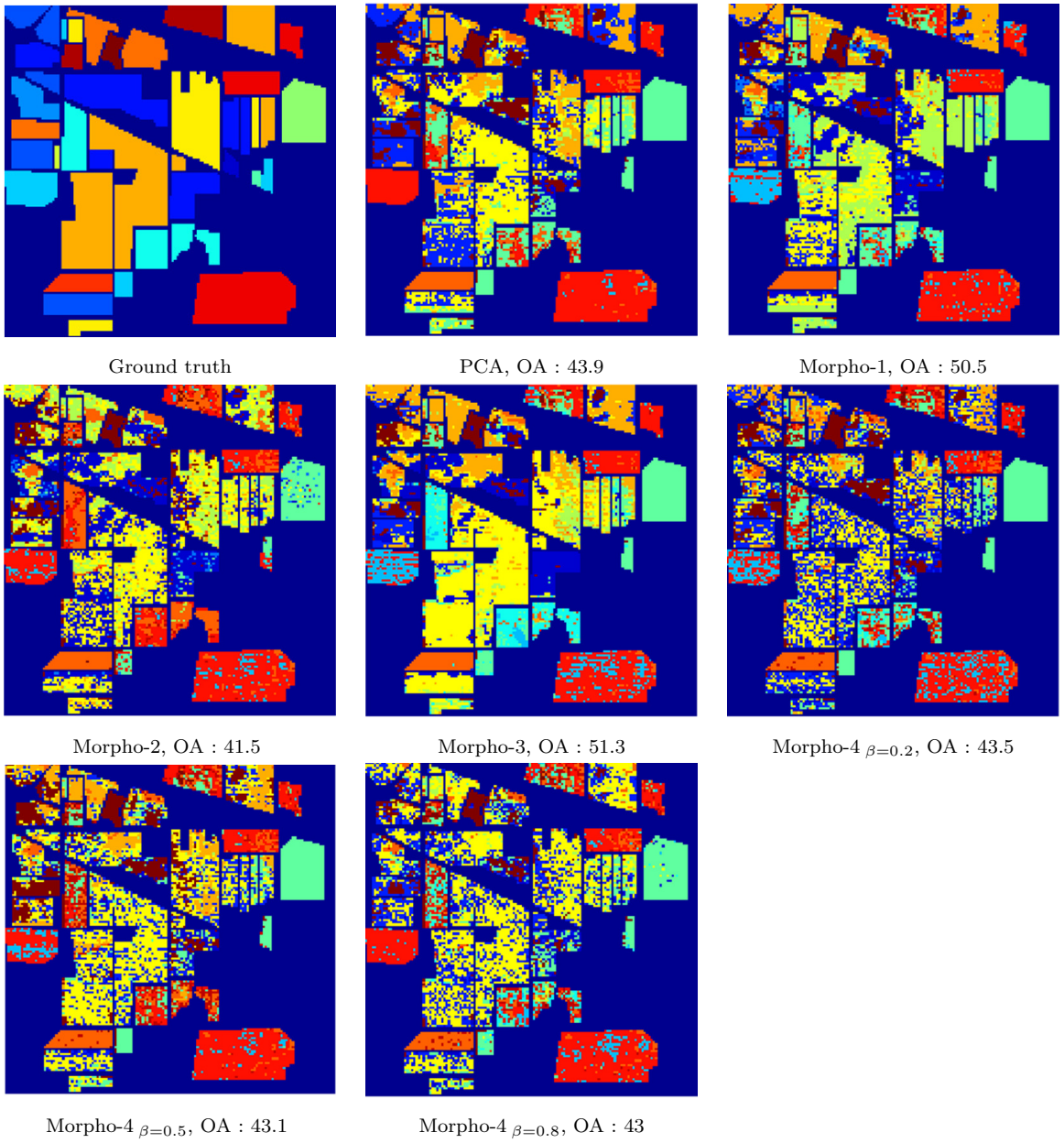


Figure 2.16: Results of supervised classification using least square SVM with a linear kernel on Indian Pines hyperspectral image. Note the OA is the overall accuracy.

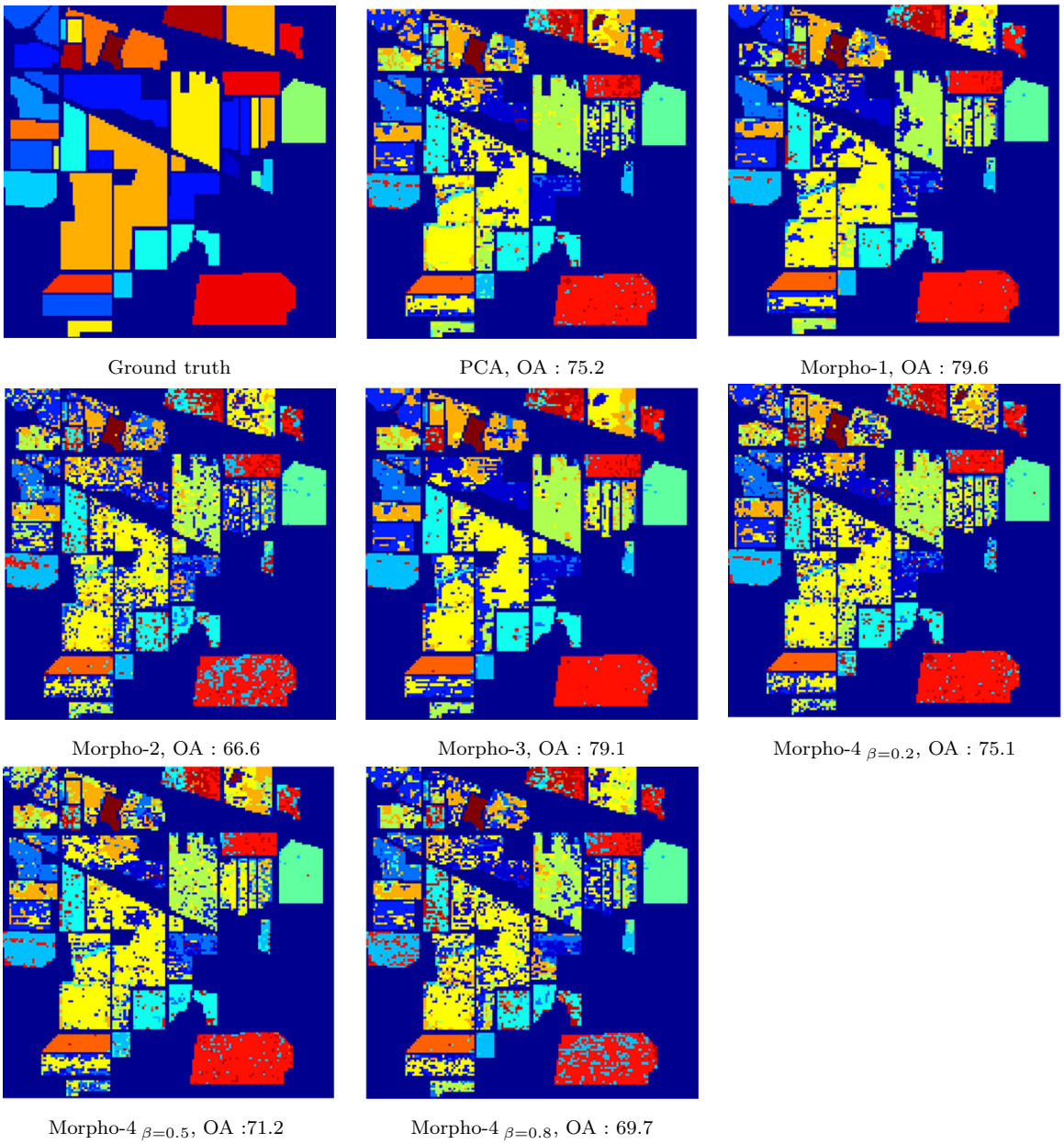


Figure 2.17: Results of supervised classification using least square SVM with a rbf kernel on Indian Pines hyperspectral image. Note the OA is the overall accuracy.

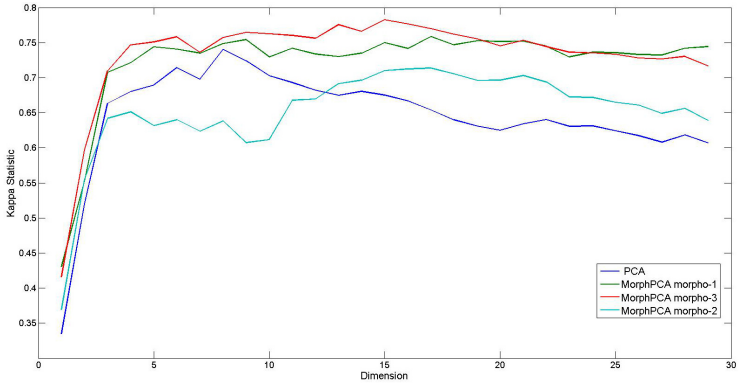


Figure 2.18: Results of kappa statistic for the least square SVM with a rbf kernel and different number of dimensions on Indian Pines hyperspectral image, the size of training set is equal to 5%.

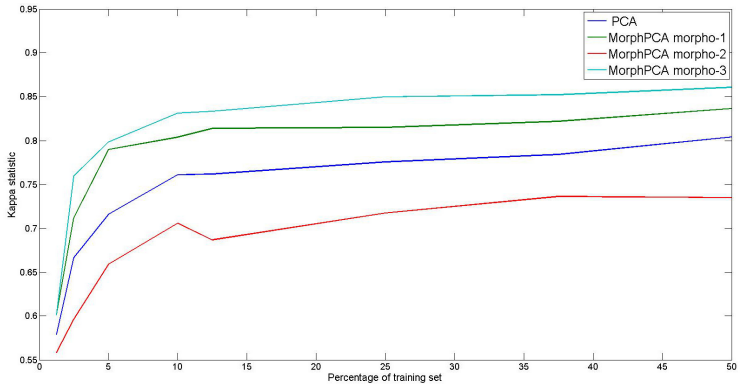


Figure 2.19: Results of kappa statistic for the least square SVM with a rbf kernel and different percentage of training set on Indian Pines hyperspectral image, the dimension of the reduced space is equal to 5.

2.5 Conclusions

We have shown in this chapter how to introduce spatial information in the process of dimensionality reduction thanks to mathematical morphology operators. The representation techniques that we introduced in the chapter are based on the notion of the MorphPCA. As we said in the introduction, it might be possible to consider for such purpose a Kernel Principal Component Analysis (KPCA), where the kernel handles jointly spatial and spectral information. However as discussed in Section 4.2, KPCA needs a Gram matrix of size $n \times n$, where n is the number of pixels. It is impossible to manipulate such a matrix with our test images on a standard computer. That is why there are some works trying to approximate the gram matrix in order to perform the KPCA. Some of them consider a small subset of the data which can be chosen randomly or according to some spatial information. This kind of approach has been used with hyperspectral images for spectral-spatial processing. But these kind of works approximate the kernel, to perform dimensionality reduction on the considered Hilbert space. Here dimensionality reduction is done on the space of the data without any approximation on the morphological covariance V_{Morpho}.

That is why we proposed techniques that are simple in computation and in memory storage and that can reduce the dimension while considering the spatial information. To assess these techniques we used typical criteria of dimensionality reduction which evaluate the fact that some properties are kept on the manifold of data after the dimensionality reduction. Moreover we also proposed some criteria to evaluate the quality of the image after the dimensionality reduction. Some of them are based on mathematical morphology, namely the 3D pattern spectrum and the $\alpha-$flat zone to check that the reconstructed image preserves global and local similarity to the original hyperspectral image. Finally, we also perform a classification of the reduced data with different techniques. According to the entire set of criteria, adding spatial information improves the dimensionality reduction. However as we can see a good dimensionality reduction is obtained when we combine spatial and spectral feature space. A technique that seems to fulfil this optimum is the MorphPCA based on the distance function.

Finally, PCA has multiple applications on image processing; typically one can use the PCA to perform denoising. For example in the case of multiple images representing the same scene but corrupted by a Gaussian noise (like on multi-temporal images), it is possible to use the PCA to reduce the dimension of the temporal data and then to project the data back on the high dimensional space, so that we reduce the influence of the noise. This process can be done with MorphPCA. Moreover another case would be to use MorphPCA on fusion of information techniques like pansharpening, which consists of increasing the spatial resolution of a multispectral or hyperspectral image thanks to a grey scale image at high resolution. Some techniques for pansharpening are based on PCA. In summary, MorphPCA can be an appropiate alternative to PCA in different image processing applications.

Chapter 3

Invariant Spatial Classification of Multi/Hyper-spectral Images

Abstract

In this chapter, a novel approach for pixel classification in multi/hyper-spectral images is proposed, leveraging on both the spatial and spectral information of the data. The introduced method lies on a recently proposed framework for learning on distributions of texture descriptors – by representing them with mean elements in reproducing kernel Hilbert spaces (RKHS). These descriptors aim at representing the content of the image while considering invariances related to the texture and to its geometric transformations, so-called spatial invariances. Moreover, we also consider spectral invariances which are related to the physical composition of the pixels. The descriptors are based on the scattering transform, which provides a useful framework for deep learning classification. Moreover, a classification algorithm learning these texture distributions is formulated and interpreted. We also study the consistency and the convergence rate of our classification technique. The performance of the theory for pixels classification is assessed on two hyperspectral images and a set of multispectral images, conventionally used. The results are good in comparison with state of the art.

Résumé

Dans ce chapitre, une nouvelle approche pour la classification des pixels d'images multi / hyper-spectral en s'appuyant à la fois sur l'information spatiale et spectrale des données. La méthode introduite repose sur un ensemble de technique récemment proposé pour pour apprendre la distribution des descripteurs de texture. Ces descripteurs visent à représenter le contenu de l'image tout en considérant les invariances liées à la texture et à ses transformations géométriques. En outre, nous considérons des invariances spectrales qui sont liées à la composition physique des pixels.

3.1 Introduction

The supervised classification of pixels in images is a difficult task, that consists in associating to each observation, which are pixels, a class that is not observed. This problem is common in remote sensing imaging, here we will apply it on multi/hyperspectral images. Hyperspectral images consist of very high-dimensional pixel observations that allows reconstruction of the spectral profiles of objects imaged thanks to the acquisition of several hundred narrow spectral bands. The supervised classification of these pixels is a challenging task, which commonly arises in remote sensing imaging [22, 59, 45, 46], but also in other domains. The structure of hyperspectral imagery imposed at first a lot of research to focus on spectral information without considering the spatial arrangement of pixels on a regular grid. The use of spatial information is a useful to fight the curse of dimensionality explained [13], and so the spectral variability of the vectors. This variability reduces the classification performance, and makes the distances between spectra meaningless. Moreover, neighboring samples might not be independent. So that if a given pixel belongs to a class, its neighbors might likely have the same class. These kinds of assumptions gave birth to the concept of regionalized random variable [100] which is a key notion of geostatistics. To be able to take into account this spatial information, we focus here on the local texture of the image. Hence, we propose a novel approach to classification able to add this spatial information in order to describe the local texture. Our technique is based on kernel embeddings of distributions and on the scattering transform and they both use both the spatial and spectral information of the data. First we present the data and we introduce the scattering transform in Section 3.3, then we provide in Section 3.4 the background on kernel embeddings of distributions, random features for fast approximations to kernel methods. In Section 3.6 we describe the kernel mean map and study the consistency and convergence rate of the proposed method and empirical evaluation is given in Section 3.9. We note that this chapter is an extended and improved version of our following contributions [54, 51].

3.1.1 Related works

Many techniques aim to include the spatial information in the image classification process. Of particular interest are those combining feature space representations describing the spatial information with those describing the pixels. Morphological feature spaces have been considered in several publications, with impressive results [35, 45, 118, 80]. On the other hand, kernel methods have also been studied extensively, and more particularly the compositions of kernels [85, 83, 21, 104], which allow building new feature space representations. We marry these approaches with a framework of [140, 109, 147, 113], where instead of the usual feature map, sending each data point to the feature space, a whole distribution can be represented in the RKHS. This idea yields a framework for learning on distributions via their representations in this RKHS. In our approach, each pixel is associated to an innovative image representation technique called the scattering transform [96, 18]. It has successfully been used to classify images [139] and sounds[5], and in this work we adapt it to be able to perform spatial pixel-wise classification. So on our approach, each pixel is described as a distribution of its neighbours of scattering transform

embedded on another RKHS. Another related line of work is that of [163], where the mean map is used on hyperspectral data to perform a dimensionality reduction. There are also recent works based on deep learning patterns [27, 93, 86], with interesting results, however they cannot compete yet against morphological features. Our proposal of deep learning technique based on texture description has the same order of classification accuracy than morphological descriptors.

Dans ce chapitre, une nouvelle approche pour la classification des pixels d'images multi / hyper-spectral en s'appuyant à la fois sur l'information spatiale et spectrale des données. La méthode introduite repose sur un ensemble de technique récemment proposé pour pour apprendre la distribution des descripteurs de texture. Ces descripteurs visent à représenter le contenu de l'image tout en considérant les invariances liées à la texture et à ses transformations géométriques. En outre, nous considérons des invariances spectrales qui sont liées à la composition physique des pixels.

3.2 Invariant classification on hyperspectral images

3.2.1 Notation

Let E be a subset of the discrete space $\mathbb{Z}^2$, which represents the support space of a 2D image and $F \subseteq \mathbb{R}^D$ be a set of pixels values in dimension D. Hence, it is assumed in our case that the value of a pixel $x \in E$ is represented by a vector $v \in F$ of dimension D. This vector v represents the spectrum at the position x, and D is the number of channels. Thus we write $f : E \to F$ the function associated to the image, such that $f(x) = v$.

3.2.2 Invariance properties of hyperspectral data

To improve the process of classification, the data should be invariant under one or more transformations underlying the "data" acquisition. For example when one want to classify the MNIST data set [78], the different letters of the data set are rotated, translated, and noised. So one would like to assign the class of images independently of the position or orientation of the characters. Such transformations may degrade the process of classification. That is why one would like to learn the invariant features of each individual class. To learn these invariants, different techniques can be used.

- One can increase the training set using variations of the training pattern according to all possible transformations. If the new training set that is synthesized is conform to the data, then the process of classification is improved.
- One can find a descriptor that may extract the invariances of each class. By extracting this feature space, the process of classification is more robust.

In this work we consider both approaches. We will first consider the second approach applied on the case of hyperspectral texture. The notion of texture [61] is linked with how human vision can interpret and recognize image structures, in

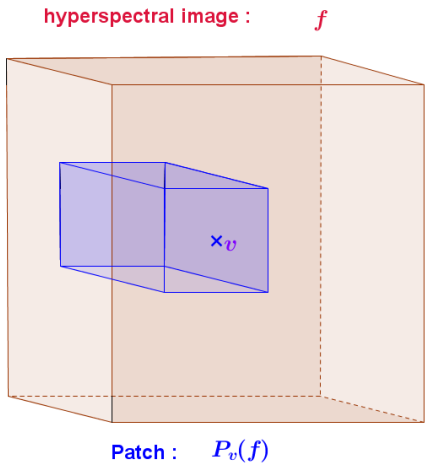


Figure 3.1: Hyperspectral patch $\mathcal{P}_v^{(s)}(f)$.

particular, how human visual system isolate and segment textures. Often natural textures keep a pattern that periodically repeat itself on a certain scale. These pixels zones have a kind of statistical periodicity. Another very important property of textures is that they generally have multiple scales. Thus the size of the neighbourhood is an important information when dealing with texture information. For our part we consider that a texture is a set of connected pixels that appears as a homogeneous area. So these pixels follow the same distribution. In particular we work with hyperspectral textures. Then, the question is how can we learn an invariant descriptor of these hyperspectral textures. We consider that image $f : E \to F$ is a random function. To classify a pixel $f(x) = v$ we will not only consider the pixel information, but also a patch around the position x, see Figure 3.1. Let us write $\mathcal{P}_v^{(s)}(f)$ a patch surrounding the pixel v, of scale s. Then we would like to classify patches, to be able to attribute the class to the central pixel v. Because of this way of classifying, there would be huge redundancy, and we would like to learn invariants to describe these patches in this context.

In our case it is necessary for hyperspectral textures that characterize pixels to be invariant to changes in illumination conditions. Therefore we would like for our descriptor to be invariant to different kind of transformations.

Spatial invariance

Let us consider two patches, $\{\mathcal{P}_{v1}(f), \mathcal{P}_{v2}(f)\}$, which can be seen as two images. Since we are first interested in the spatial invariance, we will assume that these patches are two grey scale images. These images are represented by mapping $\psi(\mathcal{P}_{v1}(f)) \in \mathcal{H}$, where $\psi(\mathcal{P}_{v1}(f))$ is a texture descriptor of the patch, and $\mathcal{H}$ is a Hilbert space that characterize texture information. As explained on [18, 96], we would like the texture descriptor to be invariant to rotation, translation, and to be Lipschitz continuous to deformation T, i.e. :

$$\|\psi(\mathcal{P}_{v1}(f)) - \psi(\mathcal{P}_{v2}(f))\|_2 \leq C\|f\|_2 \sup_x \| \bigtriangledown (T(x))\|_2, \text{ with } \mathcal{P}_{v2}(f) = T(\mathcal{P}_{v1}(f)) \quad (3.1)$$

where C is a constant, and $\bigtriangledown(T(t))$ is the deformation gradient, such that its norm 2 expresses the magnitude of the deformation. This formula expresses a stability to

deformation.

Spectral invariance

To deal with spectral invariance we use a linear model of hyperspectral images. Pixel values of a hyperspectral image can be seen as embedded into a low dimensional set in a set of materials present in the scene, that are called endmembers. The dimension of each pixel can be reduced to the number of materials, where the new barycentric coordinates of each pixel are the abundances. By abundance we mean the positive quantity of each material on the pixel. Under the linear model, each pixel of the image can be written as a nonnegative combination of the different endmembers. Thus, if we consider the spectrum at a pixel as the vector $f(x) \in \mathbb{R}^D$, then it can be written as:

$$f(x) = \sum_{r=1}^{R} a_r(x) m_r + n_x, \tag{3.2}$$

where $\{m_r\}_{r=1}^R$, $m_r \in \mathbb{R}^D$ represents the set of R endmembers, $a_r(x)$ the abundance at vector i of each endmember r, and n_i an additive noise. This last term can be neglected. Hence, the extraction of endmembers can be seen as finding the simplicial cone containing the data. In general, for a given set of vectors there are many possible simplicial cones containing the vectors [38]. A way of reducing the number of possible representations consists in restricting the nonnegative coefficients a_r to be a convex combination such that $\sum_{r=1}^R a_r(x) = 1$, $\forall x$. By projecting all the data into the simplex of the abundance, each patch is then projected on the sample simplex. Then we have for each pixel v

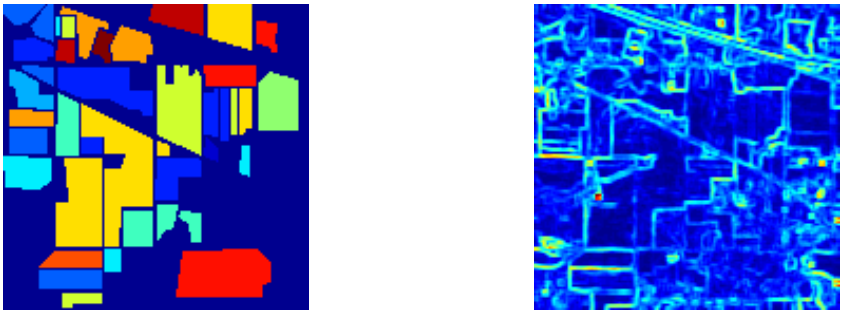
$$\mathcal{P}_v(f) \in \mathbb{R}^{n_1 \times n_2 \times D} \longrightarrow \mathcal{P}_v(a) \in \mathbb{R}^{n_1 \times n_2 \times R} \tag{3.3}$$

With $n_1 \times n_2$ being the spatial size of the patches. If we consider a second patch such that: $\mathcal{P}_{v2}(f) = \alpha.\mathcal{P}_{v1}(f)$, where α corresponds to an illumination change factor. The additional convex constraint guarantees that we are invariant to illumination changes, since $\mathcal{P}_{v2}(a) = \mathcal{P}_{v1}(a)$

3.2.3 Support vector machine classification for remote sensing

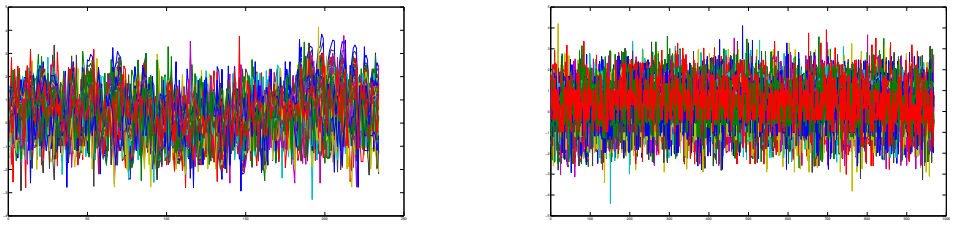
The classification of hyperspectral pixels turns out to be difficult, due to the high dimensionality of the pixels. In addition, the spectral data are noised, thus not all the sources that are present on the sample are necessary present on the acquired image. The process of image acquisition may also produce nonlinearities in the data. For all that reasons, it is considered that the data lie in a low dimensional complex manifold. The data present a high spatial and spectral variability. We represent in Figure 3.2 the ground truth and the local standard variation of the Indian Pine data set, which is a hyperspectral image of size 145×145 pixels, with $D = 224$ spectral bands. As we can see on this Figure, locally the spatial variability is not significant except at the borders of classes. Figure 3.3 gives the different pixels of the classes corresponding to corn and the soybean notill of the same hyperspectral image. As we can see, the different spectra are quite different, and finding spectral

invariants might be difficult. Because of these different factors, the classification of hyperspectral pixels is a challenging problem. To resolve this task we can use different techniques based on machine learning classification.



(a) (b)

Figure 3.2: (a) The ground truth for the "Indian Pine" dataset. (b) The local standard variation for this image, which represents the local spatial variability of the spectral image.



(a) (b)

Figure 3.3: (a) The different spectra of the class number 4 of the "Indian Pine" dataset, corresponding to "corn class". (b) The different spectra of the class number 10 of the "Indian Pine" dataset, corresponding to "Soybean-notil class".

Let us consider a set of data $\{(v_1, y_1), \ldots, (v_n, y_n)\} \in F \times \mathcal{Y}$ independent and identically distributed (i.i.d.) random variables following an unknown joint distribution $\mathcal{P}$. We call this collection of points the training set. To simplify the discussion, we will consider a two-classes $y_i \in \{-1, 1\}$ situation, the case of multiple classes can be adapted thanks to well known techniques [16]. Then, we consider that v_i is a hyperspectral spectrum, and y_i is equal to one of the classes. Let us write g: $\mathcal{X} \longrightarrow \mathbb{R}$ a prediction function, such that if $g(v_i) \geq 0$ then the class of v_i is $\hat{y}_i = 1$, and in the contrary if $g(v_i) < 0$ the class of v_i is $\hat{y}_i = -1$. The goal of the classification process is to find g such that $g(v_i)y_i \geq 0$ for most of the data of the training set. Then the goal of the process is to maximize the margin function $g(v_i)y_i$, which is the smallest distance between the decision hyperplane and any data on the training set. Trying to find the hyperplane that would fit all the data is not always optimal, so instead of using this boundary, usually we determine a hyperplane by considering that only the nearby data matters. This particular points are called the support

vectors. Then the new margins to be maximized are those of the closest data v_i from the possible hyperplane. This process of classification is called the Support Vector Machine (SVM). There are different kind of processes where the relevance of the training data set is not the same. This importance is modelled by a non increasing risk function Φ, which is a Lipschitz function. The choice of this risk function changes the algorithm, for example for the SVM the risk function is equal to $\Phi(u) = \max(1-u, 0)$. The objective function we want to minimize is :

$$\mathcal{R}(g) = \inf_{g \in \mathcal{H}} \mathbb{E}_{(v,y)\sim\mathcal{P}} \left(\Phi(g(v).y)\right). \tag{3.4}$$

Since $\mathcal{P}$ is unknown the real function we optimize is the following empirical cost function :

$$\hat{\mathcal{R}}(g) = \inf_{f \in \mathcal{H}} \frac{1}{n}. \sum_{i=1}^{n} \left(\Phi(g(v_i).y_i)\right) \tag{3.5}$$

We assumed that the training set is linearly separable, so we can find an hyperplane to separate the two classes. Unfortunately this is not always the case, that is why there have been a lot of works in classification aiming at finding the good hyperplane that would bring the data into a space in which the classes are linearly separable. Moreover, in some cases, the classes may overlap so an exact separation of the classes would lead to a poor generalization. Then the SVM formulation is modified such that it allows for some points of the training set to be on the wrong side of the margin. This new constrain added to the SVM is controlled by a parameter C, see [55] for more explanations. In the particular case of pixelwise classification of remote sensing images, an additional problem may appear: the size of the training set is small with respect to the variability of the data. To overcome this issue it is possible to generate data as proposed in [37].

3.2.4 Invariance thanks to training set generation

Learning an hyperplane to separate the data may be really tricky, and may depend on the number of available data. To encourage the generalization of the model, one can decide to generate random variation of the training set. The training set must be increased thanks to desired invariance. Then it is expected that thanks to this new training set the process will learn the invariances from the artificially enlarged training data, and will take into account them to build the hyperplan. The method of generating virtual examples must be applied carefully. As we discussed before, all the data do not have the same importance. Thus in [131], it has been decided to first proceed to a classical SVM, then learn the support vectors, and apply to this data the generation of random variation to increase the invariance. Finally another SVM is done with the new training set. This process is called the virtual SVM (VSVM). It has been applied on the MNIST data set [131], and also on remote sensing images [71]. On [71] the authors suggest some invariance properties for remote sensing. It happens that our feature space already does a patch-based classification, and is also invariant to scales. So we suggest to add Gaussian random noise to the SV, and also to increase the data set by applying some small rotation to the SV.

3.2.5 Invariance thanks to morphological profiles

Mathematical morphology operators are non-linear image processes based on the spatial structure of the image. Let f be a grey scale image which can be represented by a function. The two basic operators in morphology are the grey-level erosion and the grey-level dilatation whose definition are respectively given by [137]:

$$\varepsilon_b(f)(x) = \inf_{h\in E}\{f(x-h)-b(h)\}, \tag{3.6}$$

$$\delta_b(f)(x) = \sup_{h\in E}\{f(x-h)+b(h)\}, \tag{3.7}$$

where b is a structuring function, which introduces the effect of the operators by the geometry of its support as well as the penalizations. We note that there are just convolutions in the $(\max,+)-$ algebra and its dual algebra [7]. We consider for simplicity uniform structuring functions which are formalised by their support set or shape B, called structuring element : $b(x) = \begin{cases} 0 \text{ if } x \in B \\ -\infty \text{ otherwise} \end{cases}$. By concatenation of these two basic morphological operators it is possible to obtain more evolved filters such as the opening and the closing [137] :

$$\gamma_B(f) = \delta_B\left(\varepsilon_B(f)\right), \tag{3.8}$$

$$\varphi_B(f) = \varepsilon_B\left(\delta_B(f)\right). \tag{3.9}$$

These operators remove from f all the bright (opening) or dark (closing) structures where the structuring element B cannot fit. However they also modify the value of pixels where B fits. Thus to avoid these artefacts it has been proposed in [118] to use geodesic opening and closing. Then by considering a set $\{\gamma_R^{(i)}\}$, $i = 1...n$, of indexed geodesic openings ,and a set $\{\varphi_R^{(i)}\}$, $i = 1...n$, of indexed geodesic closings where, typically, the index i is associated to the size of the structuring element. Then thanks to the granulometry axiomatic [137] we obtain a scale-space representation of an image, which allows an image structure decomposition at different scales. Then the Morphological Profile (MP) of a grey scale image f at pixel x is defined as the $2 \times n + 1$ dimension vector:

$$MP(x) = \begin{bmatrix} \gamma_R^{(1)}(f)(x) \\ \vdots \\ \gamma_R^{(n)}(f)(x) \\ f(x) \\ \varphi_R^{(1)}(f)(x) \\ \vdots \\ \varphi_R^{(n)}(f)(x) \end{bmatrix} \tag{3.10}$$

To be able to use the MP on hyperspectral images, we first reduce the dimension of the data thanks to PCA, and then project the data on a d dimensional space which is of smaller dimension than the original space. So the hyperspectral image is represented by d grey scale eigen-images, then on each of these images we calculate the MP and we concatenate them. Hence, the spatial feature space is of dimension $d \times 2 \times n + 1$.

3.2.6 Outline of the deep weighted mean map scattering representation

We will introduce here the outline of our process of classification. We will develop in the next sections each independent step. To do the classification of the pixels of the hyperspectral images the data is projected onto a feature space. To build this feature space, we first reduce the dimension of the hyperspectral image by unmixing, which involves the computation of endmembers, and then project them into a simplex of smaller dimension whose image representation is called abundance maps. Once the data lie on a small dimensional simplex, we apply the scattering transform to the abundance maps. Instead of extracting patches and applying the scattering transform to each patch, we apply the scattering transform and keep the size of the data. This process increases the speed of calculation of the feature field. Then we use this feature space to estimate a random rbf feature space of finite dimension see section 3.4.2 for details. Thanks to this explicit feature space a weighted mean map can be computed [47]. This descriptor represents the layer 1, and which is the input for another step of random feature space computation and weighted mean pooling as input feature space the descriptor of layer 2. We can iterate it as many times as necessary, in our case of application due to the size of the training set there were no need to iterate it. Finally we concatenate the descriptors of the different layers, and apply a VSVM on it.

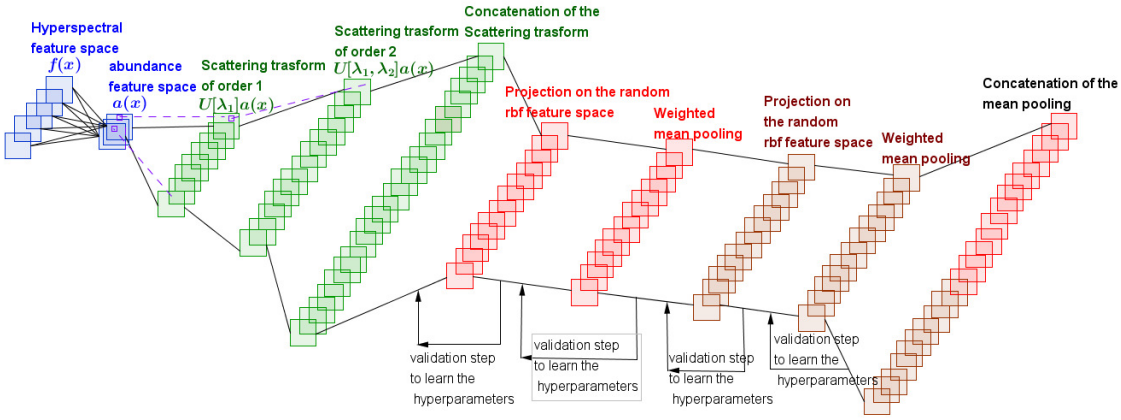


Figure 3.4: The different steps of the deep weigthed mean map scattering process .

3.3 Scattering transform

Scattering transform has been recently studied [18, 96] to describe image texture. Here we deal with the notion of local texture on digital images, so we need a space and frequency representation for discrete signals. Moreover, since we apparently do not know what is the scale of the data, a multiscale analysis of the signal is naturally required. A tool to fulfil this requirement already exists and is based on the wavelets. Let $\psi \in L^2(\mathbb{R})$ be an orthonormal wavelet, so that :

$$\int_{-\infty}^{\infty} \frac{|\hat{\psi}(x)|^2}{|x|^2} dx < \infty. \tag{3.11}$$

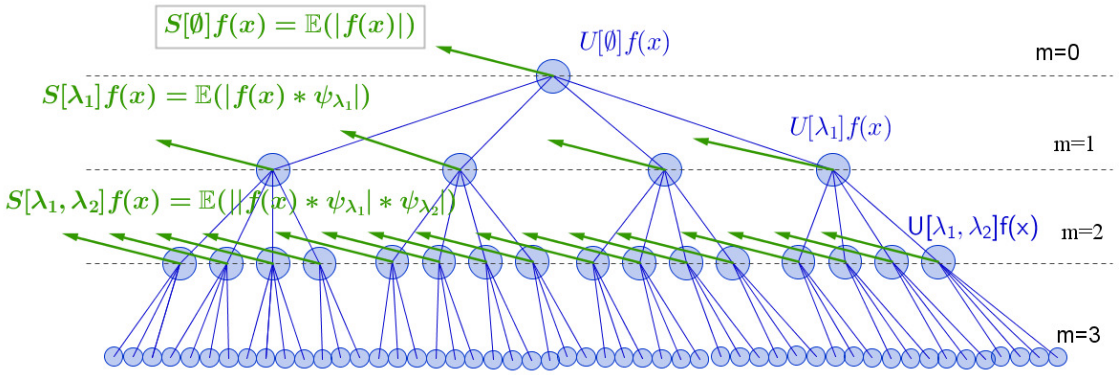


Figure 3.5: Process of scattering transform on a signal.

We can define the 1D wavelet transform as:

$$[W_\psi f](a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} \overline{\psi\left(\frac{x-b}{a}\right)} f(x)dx, \tag{3.12}$$

where the bar denotes the complex conjugate, a is the scaling, and b the time (in the case of a one dimensional signal). Wavelets are translation covariant, and so they follow the translation of the signal. If we consider two textures ν_i and $\tilde{\nu_i}$ such that $\tilde{\nu_i}(x) = \nu_i(x+h)$ then we have : $[W_\psi \tilde{\nu_i}](a,b) = [W_\psi \nu_i](a, b+h)$. The authors of [18, 96] explained that to build a translation invariant representation it is necessary to introduce a linear or non-linear operator M that commutes with translation. The operator M introduced is the mean pooling, which corresponds to a mean filter in a fixed window. Nevertheless, applying this operator to wavelets might give a translation invariant result equal to zero because of (3.11). To overcome this issue, mean pooling is applied on the modulus of the wavelets. On images, we deal with two dimensional signals, so the wavelet should be invariant to translation in all the different directions. To achieve this goal, it is proposed in [18] to use a 2D discrete wavelet transform which is a representation of 2D data according to 4 variables: dilation, rotation, and position. Let $x \in E$, if $f(x) \in L^2(\mathbb{R}^2)$ is square-integrable on E, then the 2D discrete wavelet transform is defined as :

$$[W_\psi f](j,b,\theta) = 2^{-j} \int_{-\infty}^{\infty} \overline{\psi\left(r_{-\theta}.2^{-j}.(x-b)\right)} f(x)dx, \tag{3.13}$$

where r_θ is the 2D rotation matrix. Let us denote $\hat{\psi}(\omega)$ the Fourier transform of the mother wavelet, and we write : $\hat{\psi}_{2^{-j}r_{-\theta}}(\omega) = \hat{\psi}(2^{-j}.r_{-\theta}.\omega) = \hat{\psi}_\lambda(\omega)$, where $\lambda = 2^{-j}.r_{-\theta}$ is an index related to a pair of dilation/rotation parameters. Then, the wavelet transform of f is $\{f \star \psi_\lambda(x)\}_\lambda$, which is redundant and does not have orthogonality property. Since the modulus of the wavelet transform is used, we deal with $Uf(x) = \{|f \star \psi_\lambda(x)|\}_\lambda$ and then a mean pooling is applied. So finally one has $Sf(x) = \{\|f \star \psi_\lambda(x)\|_1\}_\lambda$ which is just the norm 1 of the wavelet coefficient, and it provides information on the sparsity of the wavelet. In our case, the mean pooling is a simple average filter with a square kernel of size s. Moreover, let us write $U[\lambda]f(x) = |f \star \psi_\lambda(x)|$. Then, one issue is that all the multiscale variability of the signal is not handled by $Uf(x)$. A solution, proposed by [18, 96] is, before calculating the mean pooling, to apply a wavelet transform on the set $Uf(x)$, and

to take their modulus and their mean pooling. They iterate this operation in order to recover all the information to be represented. Thanks to this technique, the variability of the texture is scattered into different paths $p = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, where $U[p]f(x) = |\ldots|f \star \psi_{\lambda_1}(x)| \star \psi_{\lambda_2}(x)| \ldots \star \psi_{\lambda_m}(x)|$. It was shown empirically in [18], that usually one can limit itself on the second order path. Then one define two representation vectors: U which represents the scattering transform without mean pooling, and S including the mean pooling:

$$U(x) = \begin{bmatrix} f(x) \\ |f \star \psi_{\lambda_1}|(x) \\ ||f \star \psi_{\lambda_1}| \star \psi_{\lambda_2}|(x) \\ \vdots \end{bmatrix}, \; S(x) = \begin{bmatrix} \mathbb{E}(f(x)) \\ \mathbb{E}(|f \star \psi_{\lambda_1}|)(x) \\ \mathbb{E}(||f \star \psi_{\lambda_1}| \star \psi_{\lambda_2}|)(x) \\ \vdots \end{bmatrix}$$

It was shown in [96] that for appropriate wavelets, the scattering transform has the following properties:

- Contractive : $\|S(f_1) - S(f_2)\|_2 \leq \|f_1 - f_2\|_2$;
- Preserve the norms : $\|S(f_1)\|_2 = \|f_1\|_2$;
- Stable to deformation : $\|S(f_1) - S(f_2)\|_2 \leq C\|f_1\|_2 \sup_t \| \bigtriangledown (T(t))\|_2$ with $f_1 = T(f_2)$.

In particular, for the Haar wavelet and with the appropriate scaling and orientation properties, the scattering coefficients are equivalent to the SIFT descriptor [92]. We can also notice that with the Haar wavelet, the scattering transform has some links with the variogram of order 1, which is a geostatistic descriptor of spatial information. We will focus on this interpretation in the Section 5.

3.4 Deep mean map

3.4.1 Mean map kernel

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel. By Moore-Aronszajn theorem [14], there is a unique RKHS $\mathcal{H}$ of real-valued functions on $\mathcal{X}$ where

$$\langle g, k(\cdot, v)\rangle_{\mathcal{H}} = g(v), \text{ for all } g \in \mathcal{H}, v \in \mathcal{X},$$

implying that $k(.,.)$ corresponds to an inner product between features and, in particular, $k(v, v') = \langle k(\cdot, v), k(\cdot, v')\rangle_{\mathcal{H}}$. This means that $k(\cdot, v)$ can be viewed as a feature of $v \in \mathcal{X}$. For many typical choices of kernels k, the RKHS $\mathcal{H}$ is infinite-dimensional. Now, let X denote a random variable following a distribution $\mathcal{P}$. The *mean map* or the *kernel embedding* [140, 146] of $\mathcal{P}$ is defined as:

$$\mu_{\mathcal{P}} := \mathbb{E}_X[k(\cdot, X)] = \int_{\mathcal{X}} k(\cdot, v) \; \mathrm{d}\mathcal{P}(v), \tag{3.14}$$

where the expectation is over $\mathcal{H}$. For *characteristic* kernels [145], which include Gaussian rbf, Matern family and many others, this embedding is injective on the space of all probability distributions (i.e., captures information on all moments, similarly to a characteristic function). Further, if we are given two random variables,

X following the distribution $\mathcal{P}$, and Y following the distribution $\mathcal{Q}$, the inner product between the corresponding embeddings is given as

$$\langle \mu_{\mathcal{P}}, \mu_{\mathcal{Q}} \rangle_{\mathcal{H}} = \mathbb{E}_{X,Y}[k(X,Y)], \tag{3.15}$$

which is sometimes referred to as a *mean map kernel*. For a random sample $\{v_1, \ldots, v_n\}$, drawn i.i.d. from $\mathcal{P}$, we can define the empirical mean map:

$$\widehat{\mu}_{\mathcal{P}} = \frac{1}{n} \sum_{i=1}^{n} k(\cdot, v_i), \tag{3.16}$$

and for random samples $\{v_1, \ldots, v_n\}$ from $\mathcal{P}$ and $\{v'_1, \ldots, v'_m\}$ from $\mathcal{Q}$, we obtain the empirical mean map kernel:

$$\langle \widehat{\mu}_{\mathcal{P}}, \widehat{\mu}_{\mathcal{Q}} \rangle_{\mathcal{H}} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k(v_i, v'_j). \tag{3.17}$$

3.4.2 Random features for kernels

The computational and storage requirements for kernel methods on large datasets can be prohibitive in practice due to the need to compute and store the kernel matrix. If we consider a dataset of n D-dimensional observations, the storage requirements are $O(n^2)$ and the calculation takes $O(Dn^2)$ operations. A remedy developed by [121] is to approximate translation-invariant kernels in an unbiased way using a random feature representation. Namely, any translation-invariant positive definite kernel k, such that $\forall (v, v') \in \mathcal{X}^2$, $k(v, v') = \kappa(v - v')$ can be written as $k(v, v') = \mathbf{E}_{\omega \sim \Lambda} \left[\cos(\omega^\top v) \cos(\omega^\top v') + \sin(\omega^\top v) \sin(\omega^\top v') \right]$, where $\omega \in \mathbb{R}^D$ follows some distribution Λ (spectral measure of the kernel). Thus, by sampling i.i.d. vectors $\omega_1, \ldots, \omega_N$ from Λ, we can approximate kernel k by $\hat{k}$ defined by:

$$\hat{k}(v, v') = \frac{1}{N} \sum_{j=1}^{N} \left(\cos(\omega_j^\top v) \cos(\omega_j^\top v') \sin(\omega_j^\top v) \sin(\omega_j^\top v') \right),$$

so that the original feature map $k(\cdot, v)$, potentially living in an infinite-dimensional space, is approximated by an explicit $2N-$dimensional feature vector:

$$\hat{Z}(x) = \sqrt{\frac{1}{N}} \left[\cos(\omega_1^\top v), \ldots, \cos(\omega_N^\top v), \sin(\omega_1^\top v), \ldots, \sin(\omega_N^\top v) \right]^T. \tag{3.18}$$

Thus, the mean map and the mean map kernel can be estimated using these finite-dimensional representations. In this part of thesis, we will focus on Gaussian rbf kernels for which the spectral measure Λ is also Gaussian.

3.4.3 Random features mean map on hyperspectral images

Let us now turn our attention to a hyperspectral image f. Around each pixel location x_i, we consider a square patch $\mathcal{P}_{x_i}^{(s)}(f)$ of size s where we will treat the pixels as a random sample from a distribution $\mathcal{P}_i$ specific to the location x_i. Instead

of calculating the kernel between individual data points, we will calculate the kernel between these distributions. An empirical mean map kernel is thus given simply by:

$$\begin{aligned} K_{mm}(x_i, x_j) &= \langle \hat{\mu}_{P_i}, \hat{\mu}_{P_j} \rangle_{\mathcal{H}} \\ &= \frac{1}{s^2} \sum_{l_1 \in \mathcal{P}_{x_i}} \sum_{l_2 \in \mathcal{P}_{x_j}} k(f(x_{l_1}), f(x_{l_2})) \\ &\approx \frac{1}{s^2} \sum_{l_1 \in \mathcal{P}_{x_i}} \sum_{l_2 \in \mathcal{P}_{x_j}} \hat{Z}(f(x_{l_1}))^\top \hat{Z}(f(x_{l_2})), \end{aligned} \tag{3.19}$$

where $f(x)$ denotes the measurement vector at location x and in the last line we employ a random feature approximation of k.

It should be noted that there maybe outliers in a patch, which can damage the estimation of the mean. Similarly to the work of [47], we proposed to use a weighted mean map, where the weights depend on spatial/spectral information. The kernels we obtain, called convolutional kernels, have also been used in [94]. In contrast to [94], however, we will use random features expansions to explicitly represent the feature space.

More precisely, the convolutional kernel is defined as:

$$\widehat{KCN}(x_i, x_j) = \sum_{l1 \in \mathcal{P}_{x_i}} \sum_{l2 \in \mathcal{P}_{x_j}} w_{ij}(\beta, x_{l1}, x_{l2}) e^{-\frac{1}{2\sigma^2}\|f(x_{l1}) - f(x_{l2})\|_2^2}, \tag{3.20}$$

The value of the weight we selected is $w_{ij}(\beta, x_{l1}, x_{l2}) = e^{-\frac{1}{2\beta^2}\|x_{l1} - x_i\|_2^2} . e^{-\frac{1}{2\beta^2}\|x_j - x_{l2}\|_2^2}$. This formula can be represented by Figure 3.6.

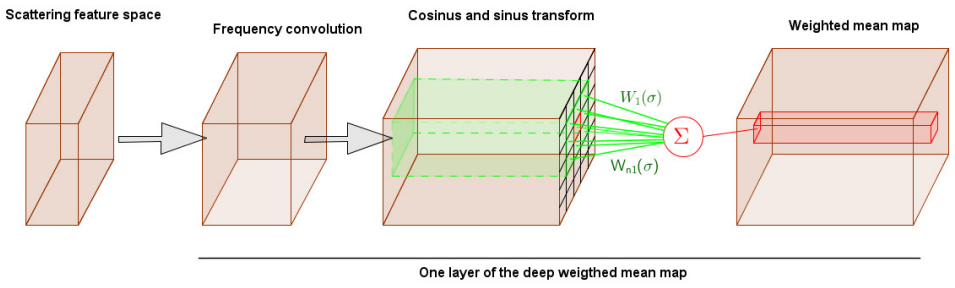


Figure 3.6: The deep weighted mean map process.

3.4.4 Weighted mean map for a spatial regularization

Let us consider a sample $\{f(x_1), \ldots, f(x_n)\} \in \mathcal{P}_{x_I}^{(s)}(f)$ drawn i.i.d. from $\mathcal{P}_I$. The empirical mean map is then:

$$\widehat{\mu}_{\mathcal{P}_I} = \frac{1}{n} \sum_{i=1}^{n} \phi(f(x_i)) \tag{3.21}$$

While $\widehat{\mu}_{\mathcal{P}_I}$ is an often used estimator of $\mu_{\mathcal{P}_I} \in \mathcal{H}$, due to the fact that the Hilbert space $\mathcal{H}$ is high- and potentially infinite dimensional whereas the number of data

on the patch is small, then Stein's phenomenon implies that it is inadmissible. To improve the performance of the estimator, the authors of [111] propose to work with a family of "shrinkage" estimators. Given that the empirical mean map is the solution of the following minimization problem:

$$\widehat{\mu}_{\mathcal{P}_I} = \text{argmin}_{g\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \|\phi(f(x_i)) - g\|^2_{\mathcal{H}}, \tag{3.22}$$

the shrinkage mean map estimator $\check{\mu}_\lambda$ solve the regularized minimization problem instead:

$$\check{\mu}_\lambda = \text{argmin}_{g\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \|\phi(f(x_i)) - g\|^2_{\mathcal{H}} + \lambda.\|g\|^2_{\mathcal{H}}. \tag{3.23}$$

Here, $\lambda > 0$ is a shrinkage parameter that balances the bias and the variance of the model. The choice of λ is explained in [110]. Finally the class of estimators proposed in [110] is :

$$\check{\mu}_\lambda = \sum_{i=1}^{n} \beta(\lambda)_i \phi(f(x_i)), \text{ with } \beta(\lambda) = g_\lambda(K)K\mathbf{1}_n, \tag{3.24}$$

with $g_\lambda(\gamma) = 1/(\lambda+\gamma)$ where γ is an eigenvalue of K. So we have:

$$\check{\mu}_\lambda = (K + n\lambda I)^{-1}K\mathbf{1}_n.(\phi(f(x_1)), ..., \phi(f(x_n)))^t. \tag{3.25}$$

This kind of shrinkage is interesting but it does not consider the spatial consistency that is really important in spatial data such as image. So to handle this problem, we propose to work with a spatial shrinkage mean map estimator $\check{\mu}^S_{\mathcal{P}_I}$ that solve the regularized minimization problem:

$$\check{\mu}^S_{\mathcal{P}_I} = \text{argmin}_{g\in\mathcal{H}} \left(\int \|\phi(f(x)) - g\|^2_{\mathcal{H}} d\mathcal{P}_I(x) dG_{\beta,I}(x) \right), \tag{3.26}$$

where $dG_{\beta,I}$ is a positive measure expressing the spatial relation between the data which are hyperspectral pixels. We decided to fix

$$dG_{\beta,I}(x) = e^{-\frac{1}{2\beta^2}\|x-x_I\|^2_2} dx.$$

The parameter β is difficult to fix by a prior knowledge, that is why we learn it during the classification. Since we do not have the distribution $\mathcal{P}_I$, we work with the empirical mean map. Therefore, we have:

$$\hat{\check{\mu}}^S_{\mathcal{P}_I} = \text{argmin}_{g\in\mathcal{H}} \left(\frac{1}{n}\sum_{i=1}^{n} \|\phi(f(x_i)) - g\|^2_{\mathcal{H}} e^{-\frac{1}{2\beta^2}\|x_i-x_I\|^2_2} \right). \tag{3.27}$$

By deriving according to g, we obtain that

$$\widehat{\mu}_{\mathcal{P}_I} = \frac{1}{\sum_{i=1}^{n} e^{-\frac{1}{2\beta^2}\|x_i-x_I\|^2_2}} \sum_{i=1}^{n} \phi(f(x_i)) e^{-\frac{1}{2\beta^2}\|x_i-x_I\|^2_2}. \tag{3.28}$$

This kind of mean map is structurally similar to the bilateral filter, that approximate the non-linear adaptive diffusion process.

3.5 Geostatistics of the feature field

Let us consider a probability space (Ω, A, P) and a domain $D \in E$. A random field or random function on the spatial domain D with values in $F \in \mathbb{R}^d$ is a function of two variables, denoted $Z(x, \omega)$. In our case, we will simplify the random field by considering that $d = 1$, which is the case that we study. For each $x_0 \in D$, $Z(x, :) : \omega \longrightarrow Z(x, \omega)$ is a random variable on (Ω, A, P). Moreover, for each $\omega_0 \in \Omega$, $Z(: \omega_0) : x \longrightarrow Z(x, \omega_0)$ is a function of $D \longrightarrow F$. We will write $Z(x)$ the random function at the position x. To study the properties of this function, one may seek to characterize stationary properties. To measure the variability of the random function $Z(x)$ at different scales, we can calculate a measure of dissimilarity between two positions x_1 and x_2. This dissimilarity between two values, designated by γ^*, is defined by:

$$\gamma^* = \frac{1}{2}(Z(x_1) - Z(x_2))^2 \tag{3.29}$$

To better characterize the fields, the dissimilarity γ^* is made dependent on the distance between the two points and orientation. By averaging dissimilarities of γ^* for all values between N_H pairs of points connected by a vector h, we get the notion of experimental variogram of order 2:

$$\gamma^*(x_i, h) = \frac{1}{2N_H} \sum_{i=1}^{N_H} \|Z(x_i) - Z(x_i + h)\|_2^2. \tag{3.30}$$

Usually, it is observed that the similarity values increase on average depending on the spatial distance of the measurement points and frequently leveled variation at huge distances. Behavior at very small scales, near the origin of the variogram, is of critical importance because it is an indicator of the degree of continuity of the regionalized variable. Furthermore, in the case of an image where it is hardly conceivable to have stationarity and isotropy properties, it seems important to limit our study of the variogram to work with low values of h. Finally, we often look at the variogram of order 2 to study the properties of a random field. But it is also quite interesting to look at the order variagram *alpha*, that bring other information, that is defined by:

$$\gamma_\alpha(h) = \frac{1}{2N_H} \sum_{i=1}^{N_H} |Z(x_i) - Z(x_i + h)|^\alpha. \tag{3.31}$$

Let us come back to our framework, with scattering transform, and especially with a Haar wavelet. We can see that the first order of the scattering transform U, at the first scale, corresponds to $\gamma^*(h = 1)$. Then to be able to deal with the variagram of order 1 we just need to calculate the local mean S. Then when we consider scattering coefficients at higher scales in a certain way we always focus on variagram of order 1 with $h = 1$, but instead of doing it on the original image, we do it on an image at a lower resolution. Then, since we decrease the resolution, each point on the random field at lower resolution is a combination of other points. So we "calculate" variogram of order 1 with more than two points. This can be seen as increasing the robustness of the calculation of the variogram, and is equivalent to increasing the

length of h. Then let us consider that $U(x)$ represents the concatenation of $\gamma^*(x,h)$ with different orientations and lengths of h.

Let us consider that we want to characterize the distribution near two positions x and y. Let us write $P_x^{(s)}(U)$, $P_y^{(s)}(U)$ the two patches extracted at two positions with $P_x^{(s)}(U) \sim \mathcal{P}_x$, and $P_y^{(s)}(U) \sim \mathcal{P}_y$. We can try to estimate the mean, the variance or other moments of $\mathcal{P}_x$ and $\mathcal{P}_y$ and compare them. Or we could try to use other estimator such as the widely used Kullback-Leibler divergence which would require the density estimation. However, since apparently we have no prior information on these distributions, we do not know if it would be sufficient. Many distances on distributions can be used. Here we focus on a innovative tool called the maximum mean discrepancy (MMD) which is defined as the squared distance between their embeddings in the RKHS [140, 144]

$$\mathrm{MMD}(\mathcal{P}_x, \mathcal{P}_y) = \left|\left|\mu_{\mathcal{P}_x} - \mu_{\mathcal{P}_y}\right|\right|^2_{\mathcal{H}}. \tag{3.32}$$

This distance is equivalent to finding the function on the RKHS that maximizes the difference in expectations between the two probability distributions, i.e.:

$$\mathrm{MMD}(\mathcal{P}_x, \mathcal{P}_y) = \sup_{||f||_{\mathcal{H}} \leq 1} \left(\mathbb{E}_X[f(P_x^{(s)}(U))] - \mathbb{E}_Y[f(P_y^{(s)}(U))]\right) \tag{3.33}$$

However, most of the time we work with a sampling of the distribution. Let us consider a random set of samples $\{v_1, \ldots, v_n\}$ from $\mathcal{P}_x$ and $\{v'_1, \ldots, v'_m\}$ from $\mathcal{P}_y$. Then we can approximate the MMD by the empirical estimate of the MMD, defined by :

$$\widehat{\mathrm{MMD}}(\mathcal{P}_x, \mathcal{P}_y) = \left|\left|\frac{1}{n}\sum_{i=1}^{n}\phi(v_i) - \frac{1}{m}\sum_{i=1}^{m}\phi(v'_i)\right|\right|^2_{\mathcal{H}} \tag{3.34}$$

$$\widehat{\mathrm{MMD}}(\mathcal{P}_x, \mathcal{P}_y) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[k(v_i, v_j) + k(v'_i, v'_j) - 2k(v_i, v'_j)\right] \tag{3.35}$$

where n and m are respectively the size of the patches $\mathcal{P}_x$ and $\mathcal{P}_y$. Let us now study the link of the MMD with characteristic functions $(\Phi_{\mathcal{P}_x}, \Phi_{\mathcal{P}_y})$ respectively of $(\mathcal{P}_x, \mathcal{P}_y)$, such that:

$$\mathrm{MMD}(\mathcal{P}_x, \mathcal{P}_y) = \left|\left|\mu_{\mathcal{P}_x} - \mu_{\mathcal{P}_y}\right|\right|^2_{\mathcal{H}}$$

$$\mathrm{MMD}(\mathcal{P}_x, \mathcal{P}_y) = \int\int\int\int \left(k(v_x, v'_x) + k(v_y, v'_y) - 2k(v_x, v_y)\right) d\mathcal{P}_x(v_x) d\mathcal{P}_x(v'_x) d\mathcal{P}_y(v_y) d\mathcal{P}_y(v'_y)$$

Since we work with positive definite and continuous kernels, we can use the Bochner theorem and rewrite the MMD as:

$$\mathrm{MMD}(\mathcal{P}_x, \mathcal{P}_y) = \int \ldots \int \left(e^{j\omega^t(v_x - v'_x)} + e^{j\omega^t(v_y - v'_y)} - 2e^{j\omega^t(v_x - v_y)}\right) \Lambda(\omega) d\mathcal{P}_x(v_x) d\mathcal{P}_x(v'_x) d\mathcal{P}_y(v_y) d\mathcal{P}_y(v'_y) d\omega$$

where Λ is the same distribution we used on the random feature space trick. Then by applying the definition of characteristic function, and their Hermitian property, we have :

$$\mathrm{MMD}(\mathcal{P}_x, \mathcal{P}_y) = \int \left(\Phi_{\mathcal{P}_x}(w)^2 + \Phi_{\mathcal{P}_y}(w)^2 - 2\Phi_{\mathcal{P}_x}(w)\Phi_{\mathcal{P}_y}(w)\right) \Lambda(\omega) d\omega,$$

so finally we obtain

$$\mathrm{MMD}(\mathcal{P}_x, \mathcal{P}_y) = \int \left(\Phi_{\mathcal{P}_x}(w) - \Phi_{\mathcal{P}_y}(w)\right)^2 \Lambda(\omega) d\omega \tag{3.36}$$

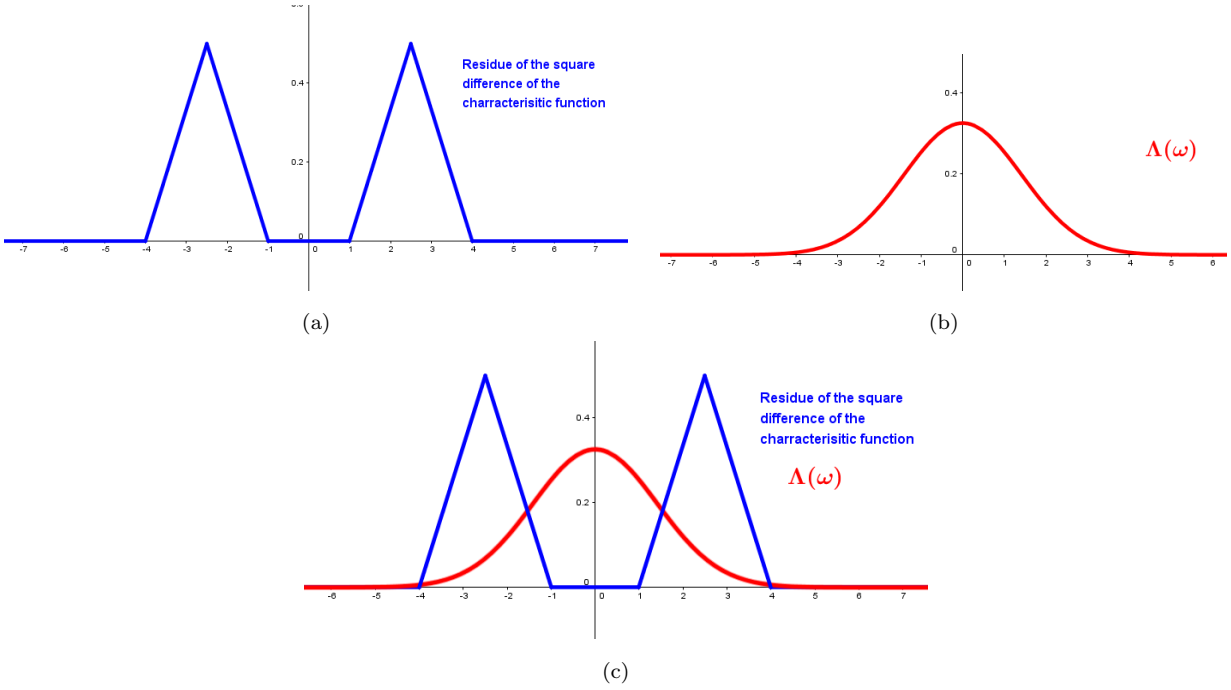


Figure 3.7: (a) The difference of two characteristic functions $\Phi_{\mathcal{P}_x}$ and $\Phi_{\mathcal{P}_y}$. (b) A possible choice of $\Lambda(\omega)$ for a possible kernel. (c) As we can see, this kernel captures the difference between the two characteristic functions.

As we can see in Figure 3.7, thanks to the MMD, we calculate the difference between two characteristic functions of the variogram that have been filtered by a low pass filter $\Lambda(\omega)$. In our case the low pass filter is a centred Gaussian function of parameter σ, that is learned during the validation step. Thanks to this technique we expect that we are able to learn well the representation of the texture according to the class we want to discriminate.

3.6 Support vector machines on kernel distribution embeddings

In images, it is natural to assume that pixels close between them are more linked than pixels fare away. Then studying region properties to find the class of pixels

seems a promising approach. We can furthermore consider that the image is a set of classes, and also that pixels are gathered in set of bags of pixels that follow the same distribution. These bags are composed of pixels close to each other. In [147] the author proposed to work with a super-pixels strategy to attribute to each super pixel the correct class. The difficulty with this kind of approach is that if the super-pixel assigned has a wrong class, then all the pixels on this super-pixel are wrongly classified. Therefore this technique depends not only on the feature extracted but also highly on how we select the super-pixels. Here we propose to use a sliding window approach, which has the advantage of a less important prior. We have calculated in [54] that if we consider a sliding window approach or a super-pixel approach and we calculate a kernel mean map SVM, then the SVM converges to a SVM between the distributions of the super-pixels.

Let us consider that the data are partitioned into sets following the same distribution, then the structure of our data is given by $\{(\{v_{i,n}\}_{n=1}^{N_i}, y_i)\}_{i=1}^{l}$ with $v_{i,1}, \ldots, v_{i,N_i} \overset{i.i.d.}{\sim} v_i$, where (v_i, y_i) are drawn from a joint meta distribution $\mathcal{M}$. We follow the notation of [147]. We represent this problem in Figure 3.8. Let us write the following expected risk function of the data for the SVM problem:

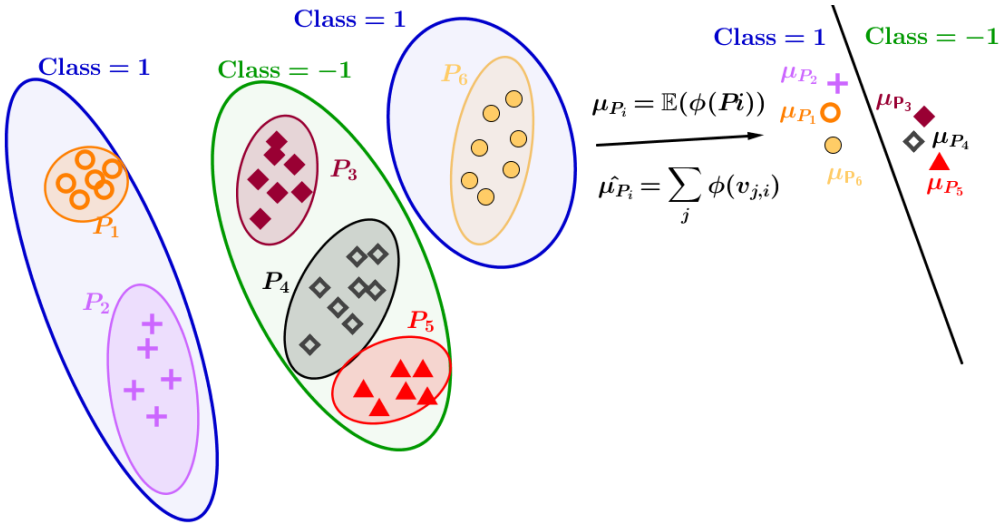


Figure 3.8: A representation of the SVM kernel embedding of the distributions, based on a finite data set partitioned into bag of distributions.

$$\mathcal{R}(f) = \inf_{f \in \mathcal{H}} \mathbb{E}_{(v,y)\sim\mathcal{M}} \left(\Phi(f(v)y)\right). \tag{3.37}$$

where Φ is a loss function. We can modify it to mean map embedding classification problem:

$$\mathcal{R}_\mu(f) = \inf_{f \in \mathcal{H}} \mathbb{E}_{(v,y)\sim\mathcal{M}} \left(\Phi(f(\mu_v)y)\right). \tag{3.38}$$

We can also write the empirical risk function, for mean map embedding classification problem:

$$\hat{\mathcal{R}}_\mu(f) = \inf_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \left(\Phi(f(\mu_{v_i})y_i)\right). \tag{3.39}$$

Finally we can also write the empirical risk function, for the empirical mean map embedding classification problem:

$$\hat{\mathcal{R}}_{\hat{\mu}}(f) = \inf_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (\Phi(f(\hat{\mu}_{v_i}) y_i)) \tag{3.40}$$

Then, we would like to obtain an inequality between $\mathcal{R}_{\mu}(f)$ and $\hat{\mathcal{R}}_{\hat{\mu}}(f)$. To do that, inspired by [109], we derive a inequality between $\mathcal{R}_{\mu}(f)$ and $\mathcal{R}(f)$.

Theorem 17 *Given that $x \sim P$ an arbitrary probability distribution with variance σ^2, a Lipschitz continuous function $f : \mathbb{R} \to \mathbb{R}$ with constant C_f, an arbitrary loss function $\Phi : \mathbb{R} \to \mathbb{R}$ that is Lipschitz continuous with constant C_l , it follows that :*

$$\mathcal{R}_{\mu}(f) - \mathcal{R}(f) \leq C_l C_f^2 \mathbb{E}_{(v)} \|v - \mu_v\|^2 \mathbb{E}_{(y)}(y^2) \tag{3.41}$$

Proof. First we have: $\mathcal{R}_{\mu}(f) - \mathcal{R}(f) \leq \mathbb{E}_{(v,y)\sim\mathcal{M}} [\Phi(f(v).y) - \Phi(f(\mu_v).y)]$
$\mathcal{R}_{\mu}(f) - \mathcal{R}(f) \leq \mathbb{E}_{(v,y)\sim\mathcal{M}} |\Phi(f(v).y) - \Phi(f(\mu_v).y)|$. Since Φ is Lipschitz continuous we obtain: $\mathcal{R}_{\mu}(f) - \mathcal{R}(f) \leq C_l \mathbb{E}_{(v,y)\sim\mathcal{M}} |f(v).y - f(\mu_v).y|$.
Thanks to the Cauchy-Schwarz inequality, we finally get:
$\mathcal{R}_{\mu}(f) - \mathcal{R}(f) \leq C_l \mathbb{E}_{(v)}(f(v) - f(\mu_v))^2 \mathbb{E}_{(y)}(y^2)$ Since f is Lipschitz continuous we have:
$\mathcal{R}_{\mu}(f) - \mathcal{R}(f) \leq C_l C_f^2 \mathbb{E}_{(v)} \|v - \mu_v\|^2 \mathbb{E}_{(y)}(y^2)$ ■

Then we might use [30] where we have an inequality between $\mathcal{R}_{\mu}(f)$ and $\hat{\mathcal{R}}_{\mu}(f)$:

Theorem 18 *Let $\mathcal{G} = \Phi(\mathcal{H}, .)$ denote the loss class, let $\mathcal{R}_n(\mathcal{G})$ denote the Rademacher complexity. Let $\Sigma(\mathcal{G})^2 = \sup_{g \in \mathcal{G}} \mathbb{E}(g^2)$ be a bound on the variance of the functions in $\mathcal{G}$. If the trace of the kernel is bounded, the loss function $\Phi : \mathbb{R} \to \mathbb{R}$ that is Lipschitz continuous, for any $\delta > 0$, the following bound holds with probability at least $1 - \delta$*

$$\hat{\mathcal{R}}_{\mu}(f) - \mathcal{R}_{\mu}(f) \leq 8\mathcal{R}_n(\mathcal{G}) + \Sigma(\mathcal{G}) \sqrt{\frac{8 \log(2/\delta)}{n}} + \frac{3 \log(2/\delta)}{n}$$

Theorem 19 *Given that $f : \mathbb{R} \to \mathbb{R}$ is a Lipschitz continuous function with constant C_f, an arbitrary loss function $\Phi : \mathbb{R} \to \mathbb{R}$ that is Lipschitz continuous with constant C_l , it follows that :*

$$\hat{\mathcal{R}}_{\hat{\mu}}(f) - \hat{\mathcal{R}}_{\mu}(f) \leq \frac{1}{n} C_l C_f^2 \hat{\mathbb{E}} \left(\|\mu_v - \hat{\mu}_v\|^2 \right) \hat{\mathbb{E}} \left((y)^2 \right)$$

Proof. $\hat{\mathcal{R}}_{\hat{\mu}}(f) - \hat{\mathcal{R}}_{\mu}(f) \leq 1/n. [\sum_{i=1}^{n} \Phi(f(\hat{\mu}_{v_i}).y_i) - \Phi(f(\mu_{v_i}).y_i)]$
$\hat{\mathcal{R}}_{\hat{\mu}}(f) - \hat{\mathcal{R}}_{\mu}(f) \leq 1/n. (\sum_{i=1}^{n} |\Phi(f(\hat{\mu}_{v_i}).y_i) - \Phi(f(\mu_{v_i}).y_i)|)$ Since Φ is Lipschitz continuous we have: $\hat{\mathcal{R}}_{\hat{\mu}}(f) - \hat{\mathcal{R}}_{\mu}(f) \leq 1/n. C_l (\sum_{i=1}^{n} |(f(\hat{\mu}_{v_i}) - f(\mu_{v_i})).y_i|)$.
Thanks to the Cauchy-Schwarz inequality we have:
$\hat{\mathcal{R}}_{\hat{\mu}}(f) - \hat{\mathcal{R}}_{\mu}(f) \leq 1/n. C_l (\sum_{i=1}^{n} |y_i|^2) (\sum_{i=1}^{n} |f(\hat{\mu}_{v_i}) - f(\mu_{v_i})|^2)$ Since f is Lipschitz continuous we obtain:
$\hat{\mathcal{R}}_{\hat{\mu}}(f) - \hat{\mathcal{R}}_{\mu}(f) \leq 1/n. C_l C_f^2 (\sum_{i=1}^{n} (y_i)^2) (\sum_{i=1}^{n} \|\hat{\mu}_{v_i} - \mu_{v_i}\|^2)$ ■

We also need the following theorem proved in [140]

Theorem 20 *Assume that $\|g\|_\infty \leq R$ for all $g \in \mathcal{H}$ with $\|g\|_{\mathcal{H}} \leq 1$, and that k is an universal kernel. Then with probability at least $1-\delta$:*

$$|\mu[P] - \mu[X]| \leq 2\mathcal{R}_n(\mathcal{H}, P) + R\sqrt{\log(1/\delta)/n}$$

where $\mathcal{R}_n(\mathcal{H}, P)$ denotes the Rademacher average associated with P and $\mathcal{H}$.

Then by combining the previous results we easily have the following theorem.

Theorem 21 *Given the conditions of the previous theorems, then with probability at least $1-\delta$, we have:*

$$\begin{aligned}\hat{\mathcal{R}}_{\hat{\mu}}(f) - \mathcal{R}(f) \leq C_l C_f^2 \Big[&\mathbb{E}_{(v)}\|v-\mu_v\|^2 \mathbb{E}_{(y)}(y^2) \\ &+ \left(2\mathcal{R}_n(\mathcal{H}, P) + R\sqrt{\log(1/\delta)/n}\right)\hat{\mathbb{E}}\left((y)^2\right)\Big] \\ &+8\mathcal{R}_n(\mathcal{G}) + \Sigma(\mathcal{G})\sqrt{\frac{8\log(2/\delta)}{n}} + \frac{3\log(2/\delta)}{n}\end{aligned}$$

This theorem states that if the random variable v is concentrated around its mean and the functions f and Φ check the conditions of the previous theorems, then the loss deviation $\hat{\mathcal{R}}_{\hat{\mu}}(f) - \mathcal{R}(f)$ will be small.

3.7 Kernel mean map scattering and invariance

As we have discuss above, it has been established in [96] that for appropriate wavelets, the scattering transform has the following properties:

- it is contractive: $\|S(f_1) - S(f_2)\|_2 \leq \|f_1 - f_2\|_2$,
- it preserves the norms: $\|S(f_1)\|_2 = \|f_1\|_2$,
- it is stable to deformations: $\|S(f_1) - S(f_2)\|_2 \leq C\|f_1\|_2 \sup_t \| \bigtriangledown (T(t))\|_2$ with $f_1 = T(f_2)$.

In our case, we perform the average pooling on the scattering coefficients $U[p]f$ that have been projected on the rbf feature space. We denote this scattering transform $S_{\mathcal{H}}[p](f)$. Then one can wonder if these properties of invariance are still satisfied. We can prove that it is still contractive, and stable to deformation, but it does not preserve the norms anymore.

Proof. Since the exponential function is convex we have that:

$$e^u \geq u + 1 \ \forall u \in \mathbb{R}$$

Thus,

$$e^{-\frac{\|v-v'\|^2}{2\sigma^2}} \geq -\frac{\|v-v'\|^2}{2\sigma^2} + 1$$

$$2 - 2e^{-\frac{\|v-v'\|^2}{2\sigma^2}} \leq \frac{\|v-v'\|^2}{\sigma^2}$$

Then, replacing by the corresponding rbf feature space we have:

$$\|\phi(v) - \phi(v')\|^2 =< \phi(v) - \phi(v'), \phi(v) - \phi(v') >$$

$\|\phi(v) - \phi(v')\|^2 =< \phi(v), \phi(v) > + < \phi(v'), \phi(v') > -2 < \phi(v), \phi(v') >= 2 - 2e^{-\frac{\|v-v'\|^2}{2\sigma^2}}$.

Hence,

$$\|\phi(v) - \phi(v')\|^2 \leq \frac{\|v - v'\|^2}{\sigma^2}$$

this means that the mapping of the rbf kernel is Lipschitz-continuous of parameter $1/\sigma^2$. Using this result we have:

$$\|S_{\mathcal{H}}(f_1) - S_{\mathcal{H}}(f_2)\|_2 = \|\mathbb{E}\left(\phi(Uf_1) - \phi(Uf_2)\right)\|_2$$

According to the fact that ϕ is Lipschitz-continuous, we obtain:

$$\|S_{\mathcal{H}}(f_1) - S_{\mathcal{H}}(f_2)\|_2 \leq 1/\sigma^2 \|S(f_1) - S(f_2)\|_2$$

Finally one gets:

$$\|S_{\mathcal{H}}(f_1) - S_{\mathcal{H}}(f_2)\|_2 \leq 1/\sigma^2 \|f_1 - f_2\|_2 \tag{3.42}$$

and

$$\|S_{\mathcal{H}}(f_1) - S_{\mathcal{H}}(f_2)\|_2 \leq C/\sigma^2 \|f_1\|_2 \sup_t \| \bigtriangledown (T(t))\|_2 \tag{3.43}$$

So the new descriptor is stable to deformation and contractive depending on a the parameter σ of the rbf kernel. ■

3.8 Multiple kernel mean map

Learning the hyperparameter of the rbf kernel is hard, and unfortunately the classification algorithm is really sensitive to this choice. That is why we first consider that $\sigma = median(\{\|v_i - v_j\|_{\mathbb{R}^D}, (i,j) \in [1,n]^2\})$ as proposed in [148]. However, it happens that this choice may not be perfectly adapted for our training set and classification algorithm. So to improve it, we split the training set into two sets. One is called the training set, and the other one the validation set. Then we learn a model on this new training set and then classify the validation set testing a family of parameters σ $\{0.01 \times \sigma, 0.1 \times \sigma, 0.5 \times \sigma, \sigma, 3 \times \sigma, 6 \times \sigma, \ldots, 30 \times \sigma\}$. After having performed a sufficient number of validation classifications, we choose the parameter that brings the best results. The issue with this technique is that the result of the choice of the parameter depends on the initial training set. In the case of remote sensing, most of the time the training set is small. To overcome this issue, we propose to build a new kernel mean map by combining a predefined set of kernels mean maps. Let us consider that we have a set of M kernels $\tilde{\mathcal{K}} = \{k_1, \ldots, k_M\}$. One can consider as a multiple kernel the sum kernel : $\mathcal{K} = \sum_{m=1}^{M} k_m$; or the weighted sum kernel : $\mathcal{K} = \sum_{m=1}^{M} \beta_m k_m$ where the coefficients β_m can be fixed or learned during the classification process. This leads to multiple kernel supervised learning techniques.

Most of these techniques [122] follow the large margin framework of the SVM: first they start with a set of training samples $\mathcal{D} = \{(v_1, y_1), \ldots, (v_n, y_n)\}$. Then, they formulate their optimization problem by as:

$$\min_{k \in \mathcal{K}} \left(\min_{f \in \mathcal{H}_{\mathcal{K}}} \left(\lambda \|f\|_{\mathcal{H}_{\mathcal{K}}} + \sum_{i=1}^{n} l(y_i, f(v_i)) \right) \right), \tag{3.44}$$

where l is a loss function, $\mathcal{K}$ is the optimization domain of the candidate kernels. There are different ways to solve this optimization problem. This kind of technique lean on the fact that we can learn the coefficient β_m thanks to a good training set. However here we will consider that we do not have enough information to learn the parameter thanks to the training set, and so we do not want to learn coefficients of the multiple kernel. We note that their are some links between the cost function of equation (3.44) and the cost function of the SVM on the sum kernels developed in [122]. In any case, we are going to use the sum kernel. We use on the one hand the fact that the linear combination of kernels is still a kernel, such that on the other hand, we use universal kernels and the linear combination of universal kernels is still a universal kernel. Then we can perform multiple kernel mean maps. We note that the use of multiple kernel mean maps has already been proposed in [58], but here we use it to classify data.

Hence, we decide to first estimate σ thanks to the training and validation sets, we write this parameter $\sigma^\star$. So we find the first kernel $k_1(v_i, v_j, \sigma^\star)$. Then we use as multiple kernel

$$\mathcal{K}(v_i, v_j) = k_1(v_i, v_j, \sigma^\star) + k_1(v_i, v_j, \gamma^{-1}\sigma^\star) + k_1(v_i, v_j, \gamma\sigma^\star),$$

where γ is a parameter that translate how much confident we are that the training set and the validation can generalize the testing set. Then we use this new kernel $\mathcal{K}$ to evaluate the mean map.

3.9 Experiments

3.9.1 Hyperspectral remote sensing

In this section, we evaluate the classification accuracy of the proposed approach using two hyperspectral images classically considered in the state-of-the-art: the AVIRIS Indian Pines data set, and the ROSIS University of Pavia data set. The first data set is an image of dimensions 145×145 pixels, with $D = 224$ spectral bands and its geometrical resolution is of 3.7 m. The dimensions of the second data set are 610×340 pixels, with $D = 103$ spectral bands and its geometrical resolution is of 1.3 m. On Pavia dataset there is a predefined testing set that we used in our experiments. In the first data set there is no common testing set so we generate 20 Monte-Carlo simulations, peaking randomly 5 pixels per class, then we aggregate the results of the supervised classification. We also generate training sets with 15 pixels per class, and with 50 pixels per class, however since there was not enough data for each class to have 50 points per class, we discarded the smallest classes. We use the C-SVM [24] as classification algorithm. We learned the parameter C thanks to a validation on the training set. Then we compared our results to the the Morphological Profile (MP)[118, 35] space which is quite common

kernel	**parameters**	**OA**	**kappa statistic**	**AA**
Linear kernel		54.6 ± 3.3	49.5 ± 3.5	58.2 ± 2.7
rbf kernel		53.9 ± 2.7	48.6 ± 3.0	58.0 ± 2.8
K_{MP}		62.9 ± 4.6	58.5 ± 5.5	66.5 ± 2.3
$K_{MP} \times \hat{Kmm}$	s=15	73.0 ± 3.7	69.7 ± 3.7	$\mathbf{76.3 \pm 2.3}$
Scattering transform U $m = 1$		73.1 ± 3.1	69.7 ± 3.4	73.7 ± 2.7
Scattering transform U $m = 2$		75.3 ± 4.7	63.5 ± 4.1	73.1 ± 4.6
Scattering transform S $m = 1$		75.2 ± 2.8	63.5 ± 2.4	73.1 ± 2.7
Scattering transform S $m = 1$	s=5	76.1 ± 3.4	69.1 ± 3.7	74.1 ± 3.8
Deep mean map scattering transform $S_{\mathcal{H}}$ $m = 1$	s=10 $\gamma = 3$	$\mathbf{77.4 \pm 3.1}$	$\mathbf{74.7 \pm 3.4}$	$\mathbf{76.2 \pm 2.5}$

Table 3.1: Overall accuracy (OA), kappa statistic, and average accuracy (AA) obtained for different kernels, applied on the AVIRIS Indian Pines hyperspectral data set. 20 Monte Carlo simulations were run. The training set is just 5 samples per class.

on pixel classification. We also use the product of the two kernels where one is the MP kernel and the other is the Kernel mean map (KMM kernel). This kind of technique has been previously explored in [85, 83, 21]. In contrast to previous works, which approximates the product of kernels thanks to addition of kernels, we can do the real multiplication since we work with finite dimension Hilbert spaces.

Let us first illustrate the intermediary image images of our algorithm using Indian Pines. Figure 3.9 shows the scattering transform U before the weighted mean map. We also represent on Figure 3.9 the weighted mean map. As one may notice, both feature spaces provide spatial data that represents the structures present on the image.

From the results on Tables 3.1, 3.2 and, 3.3 we can note that the scattering transform can compete with morphological profiles [35, 85] on Indian Pines data set. In addition, on the Pavia data set, we performed a Convolutional Neural Network (CNN) classification. CNN are mostly used to describe the texture of an image in order to classify it. Here we use CNN to classify patches. They are different CNN architectures that could be used. After trying several architectures we choose architecture inspired by the Oxfordnet [155]. The input patch is of size $32 \times 32 \times 3$, such that to have a feature space of size 3, a PCA is performed on the Pavia hyperspectral image. We apply to these patches a first convolution of size $5 \times 5 \times 3 \times 32$. Then a max pooling with sub-sampling of factor 2, followed by a Relu layer is applied. Then we perform a new convolution layer of size $5 \times 5 \times 32 \times 32$ followed by a Relu and an average pooling with sub-sampling of factor 2. We apply a convolution layer of size $5 \times 5 \times 32 \times 64$, then a Relu layer and an average pooling with sub-sampling of factor 2 , followed by a new convolution layer of size $4 \times 4 \times 64 \times 64$. Then as regularizer we apply a dropout to cancel 30% to cancel the less powerful coefficients. Then we apply two fully connected layers. Obviously, this network is probably not the most optimal one, however it gives us a hint of some results on performance of convolutional neural networks applied on our problem of pixel classification. Moreover since the training set is quite small, the training set has been increased by rotating the patch according to 8 orientations : $\{0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4\}$. Table 3.4 gives the results of the different descriptors. One can conclude that our descriptor can compete with the others.

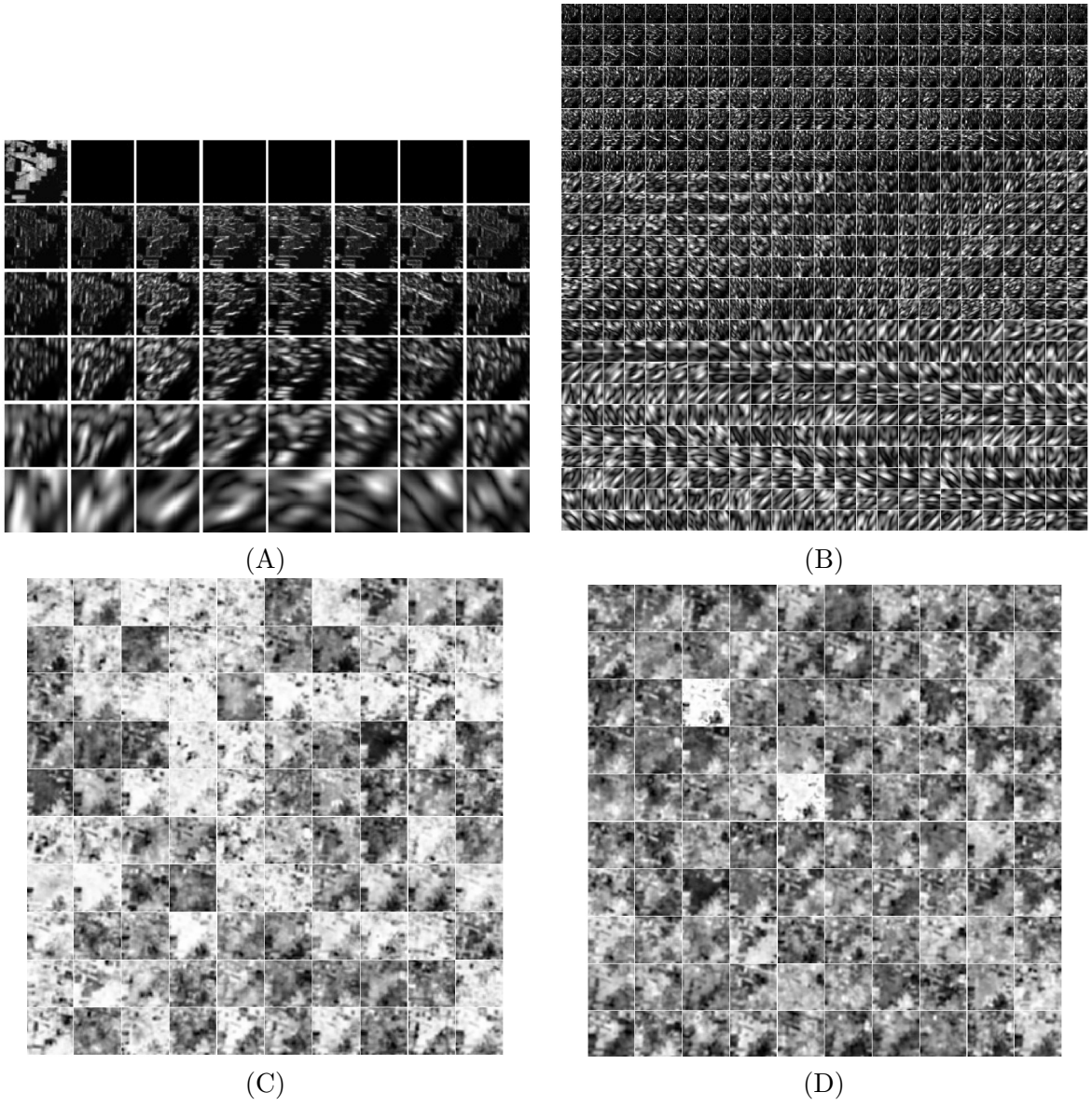


Figure 3.9: In (A) the zeros and first order scattering before the mean, U of an abondance map are represented. On the first line the abondance map is given, and then each line is composed of the 8 orientations at a given scale. In (B) we represent the second order scattering before the mean, U. In (C) and (D) we represent respectively the weigthed deep mean map first layer and second layer.

kernel	**parameters**	**OA**	**kappa statistic**	**AA**
Linear kernel		67.0 ± 1.8	63.1 ± 1.9	67.9 ± 2.3
rbf kernel		68.6 ± 2.1	65.7 ± 2.2	69.9 ± 2.3
Morphological profile		77.1 ± 1.6	74.1 ± 1.8	73.8 ± 2.4
$K_{MP} \times \hat{Kmm}$	s=15	86.7 ± 2.0	85.0 ± 2.2	85.7 ± 1.2
Scattering transform U $m = 1$		85.0 ± 2.0	83.1 ± 2.2	82.3 ± 1.6
Scattering transform U $m = 2$		86.2 ± 2.1	84.5 ± 2.4	$\mathbf{87.3 \pm 1.4}$
Scattering transform S $m = 1$	$s = 5$	87.6 ± 2.1	86.0 ± 2.4	82.6 ± 1.9
Scattering transform S $m = 2$	$s = 5$	88.0 ± 2.2	86.5 ± 2.4	83.9 ± 2.0
Deep mean map Scattering transform $S_{\mathcal{H}}$ $m = 1$	$s = 10$	$\mathbf{89.5 \pm 1.4}$	$\mathbf{88.1 \pm 1.5}$	84.6 ± 1.8

Table 3.2: Overall accuracy (OA), kappa statistic, and average accuracy (AA) obtained for different kernels, applied on the AVIRIS Indian Pines hyperspectral data set. 20 Monte Carlo simulations were run. The training set is just 15 samples per class.

kernel	**parameters**	**OA**	**kappa statistic**	**AA**
Linear kernel		75.4 ± 1.1	72.3 ± 1.2	76.1 ± 1.9
rbf kernel		77.3 ± 1.0	74.4 ± 1.1	77.1 ± 1.9
Morphological profile		85.1 ± 2.2	83.1 ± 2.4	85.6 ± 1.5
$K_{MP} \times \hat{Kmm}$	s=15	92.8 ± 0.5	91.1 ± 0.6	$\mathbf{93.0 \pm 0.6}$
Scattering transform U $m = 1$		90.27 ± 0.8	88.60 ± 1.0	90.2 ± 0.6
Scattering transform U $m = 2$		91.0 ± 1.0	89.1 ± 1.1	91.7 ± 0.6
Scattering transform S $m = 1$	$s = 5$	91.1 ± 0.9	90.2 ± 1.1	92.4 ± 0.7
Scattering transform S $m = 2$	$s = 5$	92.6 ± 0.7	91.7 ± 0.9	93.8 ± 0.5
Deep mean map Scattering transform $S_{\mathcal{H}}$ $m = 1$	$s = 10$	$\mathbf{96.2 \pm 0.5}$	$\mathbf{95.9 \pm 0.6}$	$\mathbf{96.08 \pm 0.4}$

Table 3.3: Overall accuracy (OA), kappa statistic, and average accuracy (AA) obtained for different kernels, applied on the AVIRIS Indian Pines hyperspectral data set. 20 Monte Carlo simulations were run. The training set is just 50 samples per class.

kernel	**parameters**	**OA**	**kappa statistic**	**AA**
Linear kernel		73.2	66.6	78.5
rbf kernel		75.1	67.6	82.6
Morphological profile		97.1	96.2	96.7
$K_{MP} \times \hat{Kmm}$	s=15	$\mathbf{97.4 \pm 0.6}$	$\mathbf{96.4 \pm 0.7}$	$\mathbf{97.3 \pm 0.6}$
CNN		76	74	73
Scattering transform U $m = 2$		92.27 ± 3.1	89.60 ± 2.8	94.69 ± 3.3
Scattering transform S $m = 2$	$s = 3$	95.23 ± 3.3	93.63 ± 3.2	96.38 ± 3.3
Scattering transform S $m = 2$	$s = 15$	93.53 ± 3.1	91.34 ± 3.4	93.18 ± 3.1
Deep mean map Scattering transform $S_{\mathcal{H}}$ $m = 1$	$s = 10$	96.5 ± 2.7	95.7 ± 2.8	96.2 ± 2.6

Table 3.4: Overall accuracy (OA), kappa statistic, and average accuracy (AA) obtained for different kernels, applied on the University of Pavia hyperspectral data set. The predefined training set was used.

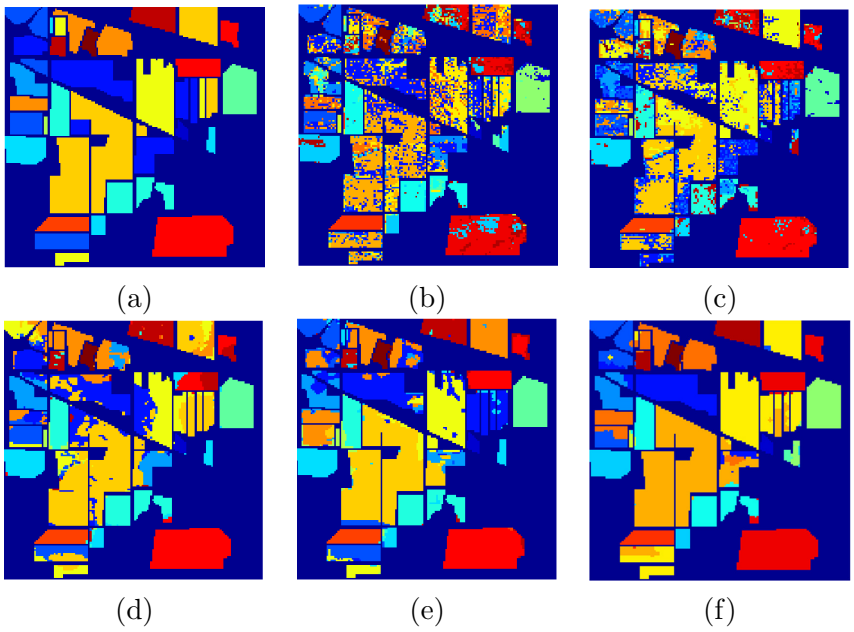


Figure 3.10: Classification maps for the Indian Pines hyperspectral image using different approaches, with just 5 points per class in the training set. In (a) ground truth, (b) the linear SVM, (c)kernel SVM with Morphological Profile, (d) kernel SVM with $K_{MP} \times K\hat{m}m$ and $s = 15$, (e) Scattering transform , (f) deep mean map Scattering transform $S_{\mathcal{H}}$ $m = 1$ and $s = 10$.

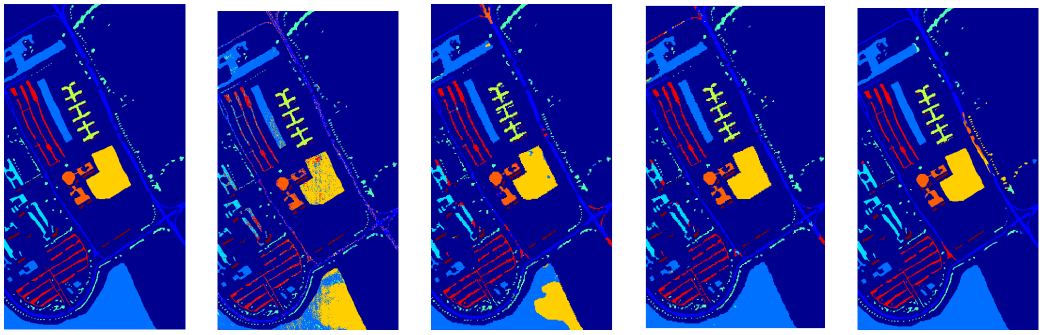
From these two data set, we criticize the independence between the training set and the testing set since they are taken on the same image, these questions are studied in the following reference [87].

3.9.2 Multispectral remote sensing

We test now our machine learning on the Zurich dataset [162]. This data set was of 20 multispectral VHR images acquired over the city of Zurich (Switzerland), that has been studied in [162]. The typical image is composed of 1.2 million pixels and they are all composed of 4 channels RGB and near-infrared (NI). The spatial resolution is about 0.61 meters / pixel. The total number of classes is 8. There are no images having the 8 classes. On the contrary to [162], we work with pixels, instead of superpixels.

We did not apply any preprocessing on the multispectral images that is composed of R,G,B and NI channels. The multiclass SVM algorithm used corresponds to the liblinear library [44] which is able to handle huge set of data with support vector classification proposed by Crammer and Singer [31]. However, because of the size of the dataset, we were not able to use the full potential of our descriptors. We just use the first order of the scattering transform with 3 rotations and 4 scales parameters, and also one layer of weighted mean map. Moreover the dimension of the estimated rbf feature space was just fixed to 30. This may degrade considerably the results of classifications.

To fix the different parameters we proceed to a validation. In order to have a training/ testing set independent, we first select 1 image that represents the testing



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

Figure 3.11: Classification maps for the Pavia hyperspectral image using different approaches, with just the classical training set. In (a) ground truth, (b) the linear SVM, (c)the estimated RBF SVM, (d) kernel SVM with KMM and $s = 15$, (e) kernel SVM with KCN and $s = 13$.

kernel	**parameters**	**OA**	**kappa statistic**	**AA**
Linear kernel		62.9 ± 3.8	57.6 ± 4.1	51.2 ± 3.9
Scattering transform U $m = 1$		65.1 ± 3.9	60.6 ± 4.2	55.1 ± 3.7
Deep mean map scattering transform $S_{\mathcal{H}}$ $m = 1$	s=3 $\gamma = 2$	80.0 ± 4.1	73.7 ± 4.3	72.2 ± 4.1

Table 3.5: Overall accuracy (OA), kappa statistic, and average accuracy (AA)obtained for different kernels, applied on the Zurich data set. 20 Monte Carlo simulations were run. The training set is composed of 19 images and the testing set of 1 image.

set. Then, on the other 19 images, we perform a validation by selecting randomly 3 images on the validation set and 16 on the training set such that on the training set we have the 8 classes. Once the parameters are learned, we train on the 19 images and then classify on the testing set. The results of the different algorithms are given on Table 3.5. As one may see, our descriptor produces relevant results in comparison with the others.

3.10 Conclusion

In this chapter we dealt with a hyper/multi-spectral pixel image classification in the context of remote sensing. We proposed a deep spatial and spectral descriptor that is able to learn the representation of the local texture of multivariate remote sensing images. Our deep descriptor has been assessed on 3 different data sets where the results were quite good. We evaluate the asymptotic properties of learning the local distribution thanks to this descriptor and a margin classifier. We compare this descriptor with different fields related to spatial statistical learning. This descriptor computes a translation invariant representation of the texture which is also Lipschitz stable to deformations. To evaluate this descriptor we just calculate the mean of the scattering coefficients provided by each layer embedded on the Hilbert space, of a rbf kernel. This links with mean map embedding. This classifier simplifies convolutional neural network since less parameters are needed. This can be an interesting way to

perform deep learning when the training data set is limited. The proposed approach improves the classification of pixel, in the case of remote sensing. Moreover it could be interesting to see its results on other data sets. Another way to improve this technique might be to put more parameters when we perform the average weighted. This might be linked to the Perceptron neural network algorithm. We could also work on simulation of data. This might be really interesting and might be considered carefully.

Part III

Fusion of information for multimodal SEM images

Chapter 4

Multimodal Scanning Electron Microscopy Images

4.1 Background on Scanning Electron Microscope (SEM)

The Scanning Electron Microscopy (SEM)[117] (see Figure 4.1) is an electron microscopy technique capable of producing high resolution images of the surface of a sample.

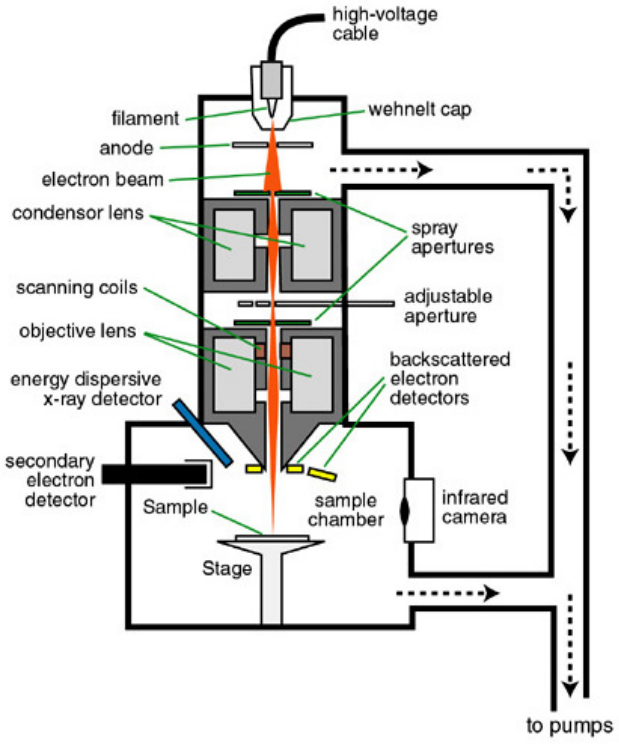


Figure 4.1: Scanning Electron Microscope overview [3].

This kind of microscope uses an extremely fine electron beam, which is produced by thermoelectronic effect from a tungsten filament; this beam is focused at a point

of the sample using an electromagnetic field. This field is generated by the coil of a condenser lens. It can, not only focus the electron beam at a single point called "cross-over", but it can also move it so that it runs through the whole sample. It is essential that the microscope column, where the electron beam spreads, remains under vacuum to avoid any deviations which could distort the measurements. Once the electron beam hits the sample, an electron-matter interaction takes place. This interaction can give birth to several types of images, and the corresponding images are shown in Figure 4.2. For our part we will focus primarily on three types of interactions:

- Backscattered electrons;
- Secondary electrons;
- X-ray.

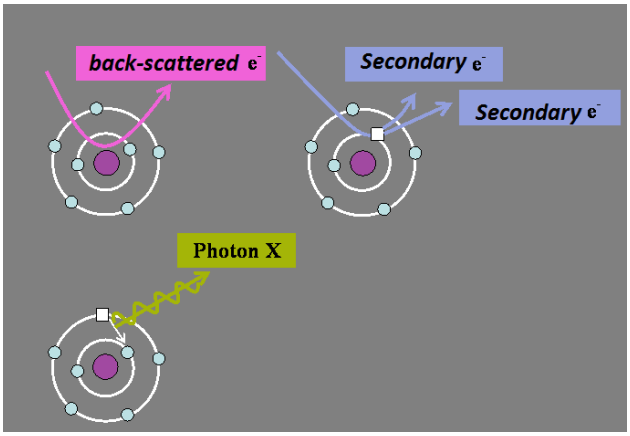


Figure 4.2: SEM interations [117].

4.1.1 Backscattered electrons

When the electron beam is sent in the direction of the sample, the electrons being negatively charged, they interact with the positively charged nucleus of the sample. Some of the electrons are re-emitted in a direction close to the incident direction. The sensor receiving electrons will recover high-energy electrons up to 30 keV. The amount of re-emitted electrons depends directly on the atomic number of the atoms constituting the sample. To capture the backscattered electrons, different types of sensors can be used. For example (annular semiconductors diode see Figure 4.3) axial devices are often used. In the case of semiconductors, they are divided into four quadrants A, B, C and D.

Two types of signals are conventionally used. First, the sum of the signals (A + B + C + D), which represents an average of the emission in a high solid angle. Under these conditions, we can interpret the image as a uniform light around the direction of the incident beam and as an almost purely chemical contrast. On a polished surface, the sensitivity is sufficient to detect small variations. The other kind of signal conventionally used is the signal difference between quadrants (AC, BD), the

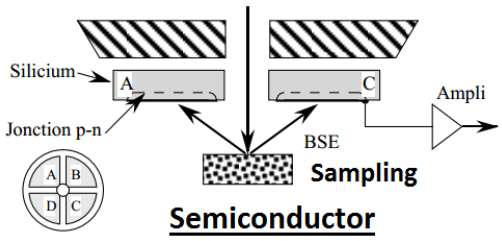


Figure 4.3: Backscattered electrons sensors [1].

small differences in function of the backscatter emission angle are amplified. This mode enhances the topographic contrast.

This is why this kind of images are used to perform homogeneity analysis. The resolution that this modality of image can reach is from one micrometer, to one tenth of a micrometer.

4.1.2 Secondary electrons

When part of the electrons of the electron beam reaches at the sample, a collision between the electron beam and the electrons of the sample happens. It follows that some of the electrons from the beam may transfer some of their energy to "external" electrons from the sample. By "external" we mean the farther electrons from the nucleus. This supply of energy to the "external" electrons causes their ejections of the atom. These electrons are called secondary electrons. Generally these electrons have low energy (about 50 eV) when they are ejected. Therefore, they cannot go very far and remain stuck into the surface of the sample analysed. The secondary electrons are collected with a positive electric field on a scintillator. The secondary electron efficiency received is dependent not only on the atomic number of the element observed, but also and especially on the angle between the incident beam and the sample observed. So in this type of image, the more enlightened areas correspond to areas of high efficiency, and the poor enlightened areas correspond to poor efficiency zones, one can speak of shadow area.

4.1.3 X-ray

The analysed sample is bombarded with a high energy electron beam about 10-40 keV. These electrons have a high energy, then they can penetrate the electron shell of atoms in the sample and excite an electron of the inner shells. The energy transfer from the electron beam to the electron in the inner shell may provoke its ejection from the atom. The ejected electron forms an electronic hole which is filled with electrons from higher energy levels. Indeed electrons from higher energy shell will gradually level down to electronic stabilization of the atom. The excitation energy of the electronic structure occurs with emission of X-ray photon, the number of X-ray photons and energy can be measured by an Energy-Dispersive Spectrometer (EDS). It is possible to characterize the different energy shells with the energy of X-ray photons and then, it is possible to characterize the electronic structure of the studied atoms. Thus we have access to information regarding the physical composition of the image.

However there are two main consequences:

- Spatial resolution of the analysis is about a micron;
- The X-ray photons emitted come from a "pear" of radius and depth at about one micron (see Figure 4.4). This may pose particular problems for the analysis of small particles or complex mixtures.

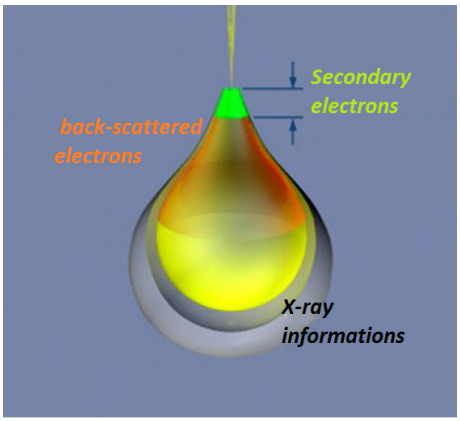


Figure 4.4: Emission domains of various interactions[117].

4.2 Noise on SEM images

As we explained before, SEM images result from X-ray photons or electrons that are detected on various captors. Let us consider now that the level of gray of a pixel is dependent on the number of photons or electrons that the sensor receives in time interval τ, and let us write dt a smaller time interval, such as $dt = \tau/K$, where $K \in \mathbb{N}$. In a way we are quantifying the time interval τ. Let us consider $< N(t, t+dt) >$ the mean number of particles (= photons or electrons) that a sensor receives between $[t, t+dt]$, ϕ the flux of particles received such that $< N(t, t+dt) >= \phi\tau/K$. We will make a first hypothesis that dt is sufficiently small so that in a time interval $[t, t+dt]$ there are just two cases:

- the sensor receives zero particles,
- the sensor receives one particle,

We will write $P(0) = 1 - q$ and $P(1) = q$ respectively the probability that the sensor in a time interval $[t, t+dt]$ receives zero particles, and one particle. Then, as one may notice, it is a binomial law of parameter q, so $< N(t, t+dt) >= q$. If we write the characteristic function of this law, we have:

$$\Psi_{dt}(\nu) = 1 - q(1 - e^{i\nu}). \tag{4.1}$$

Now we have the characteristic function for a time interval of size dt, but we would like to have the characteristic function over τ. To obtain it, a second hypothesis is needed. We consider that the K intervals are independent, such that :

$$\Psi_\tau = (\Psi_{dt}(\nu))^K = \left(1 - q(1 - e^{i\nu})\right)^K. \tag{4.2}$$

Then we consider the asymptotic case so $K \to \infty$, such that:

$$\Psi_\tau = \lim_{K\to\infty} \left(1 - \phi\tau/K(1 - e^{i\nu})\right)^K, \tag{4.3}$$

and finally we have:

$$\Psi_\tau = e^{\left(-\phi\tau(1-e^{i\nu})\right)} \tag{4.4}$$

Therefore it is possible to recognize the characteristic function of a Poisson law of parameter $\phi\tau$. Thus the number of particles in a interval τ follows a Poisson distribution of parameter $\phi\tau$ that can be written:

$$P(n) = e^{-\phi\tau}(\phi\tau)^n/n!.$$

So contrary to hyperspectral images, where a classical assumption is that these data are corrupted by an additive Gaussian noise, here EDS data are corrupted by a Poisson noise. The processing on hyperspectral images may not necessarily work with EDS images. That is why we advice to do a variance stabilization techniques. These are techniques aiming at applying a transform Φ over the entire noisy image f corrupted by a Poisson noise so that the distribution of each pixel of the converted image, noted $\Phi(f)$, is approximately Gaussian.

A variance stabilization technique conventionally used is the Anscombe transform [8]. The transform applied to the image is:

$$\Phi(f(x)) = 2\sqrt{(f(x) + \frac{3}{8})} \tag{4.5}$$

Not only Anscomb transform stabilizes the variance of the data to 1 whatever the average value was, but also this transform will turn the data distribution to Gaussian one. After stabilizing the variance, we can perform our conventional processing, and finally we return to the data in the original space by a reverse Anscombe transform

$$\Phi^{-1}(y) = \left(\frac{y}{2}\right)^2 - \frac{3}{8} \tag{4.6}$$

Chapter 5

Enhanced EDX images by fusion of multimodal SEM images using pansharpening techniques

Abstract

The goal of this chapter is to explore the potential interest of image fusion in the context of multimodal scanning electron microscope (SEM) imaging. In particular, we aim at merging the backscattered electron images that usually have a high spatial resolution but do not provide enough discriminative information to physically classify the nature of the sample, with energy-dispersive X-ray spectroscopy (EDX) images that have discriminative information but a lower spatial resolution. The produced images are named enhanced EDX. To achieve this goal, we have compared the results obtained with classical pansharpening techniques for image fusion with original approaches tailored for multimodal SEM fusion of information. Quantitative assessment is obtained by means of two SEM images and a simulated dataset produced by a software based on PENELOPE.

Résumé

Le microscope électronique à balayage (MEB) permet d'acquérir des images à partir d'un échantillon donné en utilisant différentes modalités. Le but de ce chapitre est d'analyser l'intérêt de la fusion de l'information pour améliorer les images acquises par MEB. Nous avons mis en oeuvre différentes techniques de fusion de l'information des images, basées en particulier sur des operateurs de mophologie mathematique ainsi que le filtre bilateral. Ces solutions ont été testées sur quelques jeux de données réelles et simulées par un logiciel basé sur PENELOPE.

5.1 Introduction

Scanning electron microscope (SEM) is a versatile imaging tool that allows to acquire images with various detectors. Images formed by secondary electrons (SE) reveals mainly topographic contrasts, images formed by backscattered electrons (BSE) in-

dicates local mean atomic number, whereas X-ray maps contains local elemental composition. Coupling a SEM with an energy dispersive spectrometer (EDS) leads to so called energy-dispersive X-ray spectrometry in the SEM (SEM-EDX). If the full X-ray spectrum is recorded on each scanned pixel, SEM-EDX produces spectral images. Typical SEM-EDX spectra are few millions pixels times few thousands of energy channels. The set of the images produced by the different SEM detectors can be seen as a multimodal image. Then, the various images can be processed independently or in a combined way. It seems also natural to consider that an improved processing would be obtained by combining the information present in the different modalities; obviously, that is true in the case where the modalities are "compatible". This process of image combination can be seen as a pratical case of the theory of information fusion.

More generally, the fusion of information can be seen either as the search for optimal representation including all relevant information sources [119] or as the search for algorithms making use of information from different modalities [164].

In the context of multimodal SEM, we decided to focus on a specific problem. We aim at merging the backscattered electron images that usually have a high spatial resolution, a good signal to noise ratio but do not provide enough discriminative information to physically classify the nature of the sample, with energy-dispersive X-ray spectroscopy images that have discriminative information but a lower spatial resolution and signal to noise ratio.

This problem is similar in some ways to the so-called pansharpening [75], which is well known in colour, multispectral and hyperspectral imaging. However there are also some significant differences. The first one is that in the case of classical image pansharpening, the panchromatic image (i.e., the image at the nominal spatial resolution) has a good correlation with the colour or multi/hyperspectral image. Even better, in some cases the "panchromatic image" is contained partly at some wavelengths (or linear combination of them) from the multi/hyperspectral one. In our case, the information between the two SEM modalities that are considered is not basically correlated. Moreover, we work with abundance maps extracted from the energy-dispersive spectroscopy images, since the raw EDS spectral image are of very high dimension and of very sparse nature.

As usual in multimodal SEM, we work with images perfectly registered. Although these images are spatially registered, it can be noticed that there are a number of potential artefacts which can appear when merging their underlying information sources. The origin of these artefacts is the fact that information can be structurally different, in the sense of the values of intensities, local contrast, presence of contours, etc. can be different between backscattered electron images and energy-dispersive spectroscopy. All these phenomena may degrade the quality of the fusion. A detailed list of the potential artefacts classicaly considered in image pansharpening can be found in [150]. It seems essential to deal with these problems in order to produce image fusion without major artifacts.

Up to the best of our knowledge, the literature on SEM image fusion is inexistent. Thus to initiate our work, we decided to take inspiration from the work of fusion of information on multi/hyperspectral imaging which is much more abundant and especially on pansharpening techniques [161, 91]. The rest of the paper is organized as follows. After providing a description of the SEM dataset used and a summary of the most frequently used pansharpening algorithms, we introduce a new

approach grounded on the bilateral filtering framework, which has been conceived in the particular case of SEM images fusion. An extensive quantitative assessment of the different algorithms is achieved to motivate discussion and conclusions.

5.2 Materials and methods

5.2.1 Multimodal SEM imaging.

SEM is able to produce high resolution images from the surface of a sample by means of an extremely small electron beam, which is focused at a point of the sample using the electromagnetic field of an objective lens [125]. When electrons of the beam hit the specimen surface, electron-matter interactions may produce secondary particles that are detected by adequate sensors. These secondary particles can produce several types of images, as shown in Figure 1. In this work we focus primarily on two types of images :

- Images formed by backscattered electrons (BSE);
- Images formed by electron-induced X-ray detected by EDS

Hence, we do not consider the Secondary electrons image, which is quite classic in SEM, due to the fact that it is less correlated with the two others modalities BSE and EDX. The different modalities acquired by the SEM provide different physical information about the sample, and consequently it can be interesting to merge them. This principle corresponds to the idea of SEM image information fusion.

5.2.2 SEM datasets.

SEM fusion methods discussed in this paper has been assessed using two real datasets and a simulated one.

The first dataset is composed of an EDX image of size 1024×768 pixels and 2024 levels of energy. Thus the image is a data cube $1024 \times 768 \times 2024$ pixels. As an EDX image, it is naturally corrupted by a Poisson process and of extremely sparse nature, i.e., many energy levels are zero. This image is from a sample composed of iron, copper, aluminium and oxygen. This image has been acquired with a Zeiss Supra 40 SEM fitted with a 10mm^2 Brucker X-Flash 4010 EDS spectrometer. Beam energy was 15keV, probe current 0.75nA and dwell time 912μs resulting in a 717s long acquisition time. To estimate the abundance map of each of these physical elements, a Gaussian model near each peak is fitted as EDS peak shapes are very close to a Gaussian shape due to electron-hole formation statistics in the detector [134, 11]. Since there are 6 detectable peaks at 6 different energies, 6 abundance maps are obtained. In the following of the paper, the multivariate image of physical abundances is called the "multispectral image". In addition to that, the BSE image of the same sample at the same spatial resolution is also available. To downscale the abundance maps, the experimental protocol that we followed is similar to the one proposed by the authors of [91]. We have degraded the multispectral image by applying a Gaussian blur, then we have downsampled this image by a factor $s = 5$. The second dataset is composed of an EDX image of size 1024×704 pixels and 2024

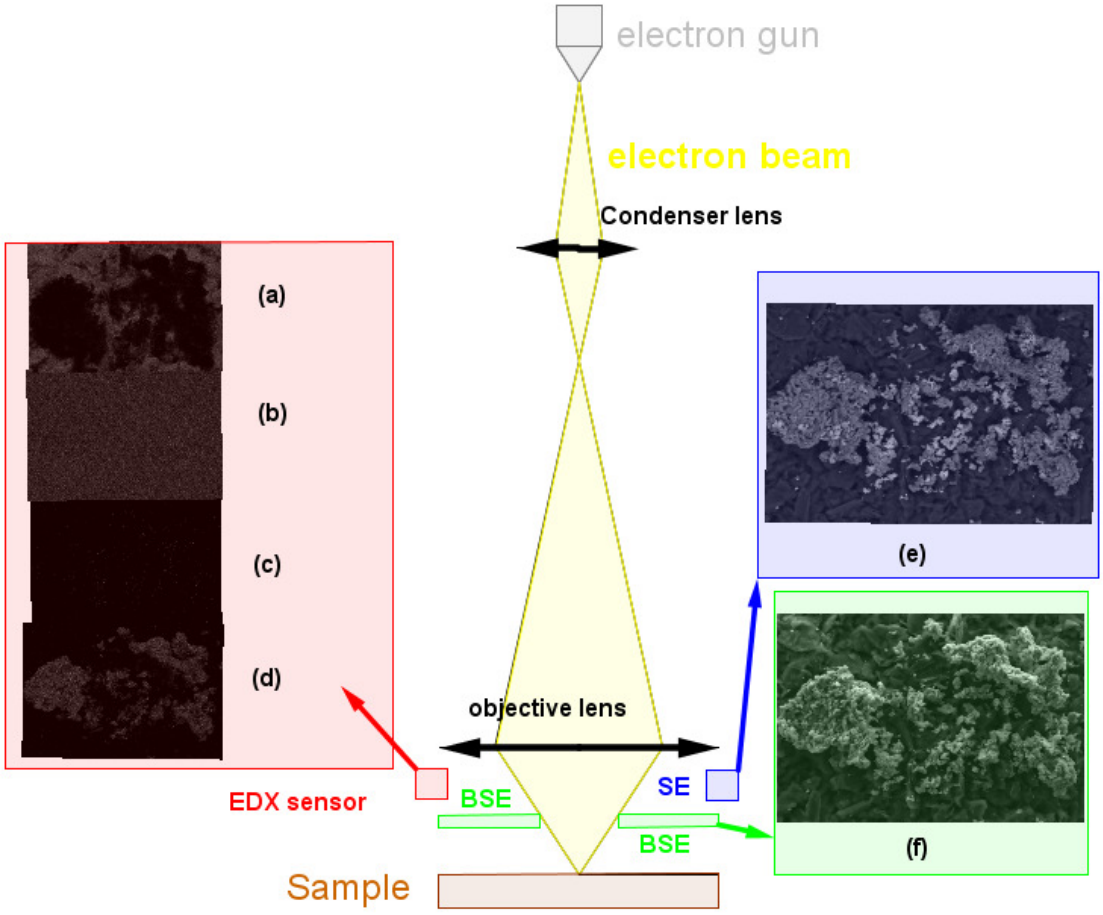


Figure 5.1: Multimodal SEM image acquisition. In the red square (a,b,c,d) we represented 4 abundances. In the blue square (e) we represented the backscattered image and in the green square (f) the secondary electron image.

levels of energy and the corresponding BSE image. This image has been acquired with a FEI Nova nanoSEM fitted with a 80mm^2 Oxford Instruments X-Max EDS spectrometer. Beam energy was 10keV, probe current 0.8nA and dwell time 3.2ms per pixel resulting in a 2307s long acquisition time. This image is from a sample composed of aluminum, oxygen, vanadium and nickel. In this case, 4 abundance maps were obtained using the same model and algorithm. Downscaled images of the abundance maps image were produced too.

In order to assess the accuracy of the algorithms, EDX images have been simulated by the Monte-Carlo method by a dedicated software based on the PENELOPE package [135, 136, 128]. First, the response of the EDS detector of the Nova NanoSEM has been characterized by measuring the full width at half maximum (FWHM) for different peak energies. The dependence on peak energy E of the standard deviation of the Gaussian peaks $\sigma(E)$ was obtained by least-square fitting and gave :

$$\sigma(E) = \sqrt{0.4403E + 337.04}, \tag{5.1}$$

with E in eV. The efficiency $\epsilon(E)$ of the detector was modeled from its geometry following the same approach proposed by Limandri et al. [89]. Analog electrons trajectories ($C1 = C2 = 0; WCC = WCR = 0$) were generated with the PENE-

LOPE package. As X-ray emission is an unusual process, both characteristic and Bremsstrahlung photon emission were enhanced by interaction forcing by a factor F with the help of the build-in functions of PENELOPE. Sample and detector geometries were handled by the PENGEOM package. The detector is annular at a 35° take-off angle to the surface. Any photon hitting the detector was considered detected with a probability $\epsilon(E)$. Each detected photon of energy E was registered in an energy channel distributed following a Gaussian law of average E and standard deviation $\sigma(E)$. Backscattered electrons were also registered to obtain a simulated BSE image. The image simulations were performed with 10000 electrons trajectories for each pixel with varying image size and forcing factor F. Varying F allowed to tune the intensity of Poisson noise of the simulated images. Two images of 1024×1024 pixels with $F = 1$ and $F = 100$ and three images of 256×256 pixels with $F = 1$, $F = 10$ and $F = 100$ were simulated. Typical simulation time was about 10 days for a 1024×1024 pixels image on an 8 processors working station. Figure 5.2 depicts the ground truth sample used in the simulation: it is composed of regular geometric shapes of pure and binary composition with low mean atomic number element (Al_2O_3, SiO_2), medium (Fe, Co, Ni) and high (Pt). Figure 5.3 provides the simulated total backscattered electrons image. It is worth noticing that phases with close mean atomic number (Al_2O_3 and SiO_2; Fe, Co and Ni) are hard to be distinguished in the BSE image. In Figure 5.4(a) is given the pixelwise mean image of the simulated EDX map. We can notice that the pixelwise mean spectrum reveals clearly the Pt zone due to high Bremsstrahlung emission in this high mean atomic number zone. Figure 5.4(b) shows the image corresponding to the Al Kα energy channel, the one having the less Poisson noise.

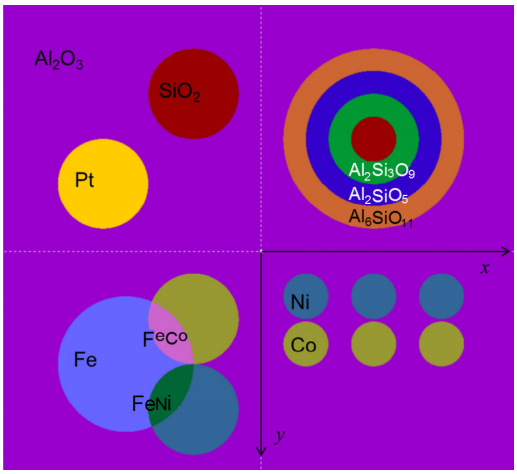


Figure 5.2: Ground truth of the simulated multimodal SEM image.

5.3 State-of-the-art

Before presenting the different image fusion methods, let us introduce the notation used in the following, which is inspired from [91].

First, from a mathematical viewpoint, a multispectral image (abundance maps)

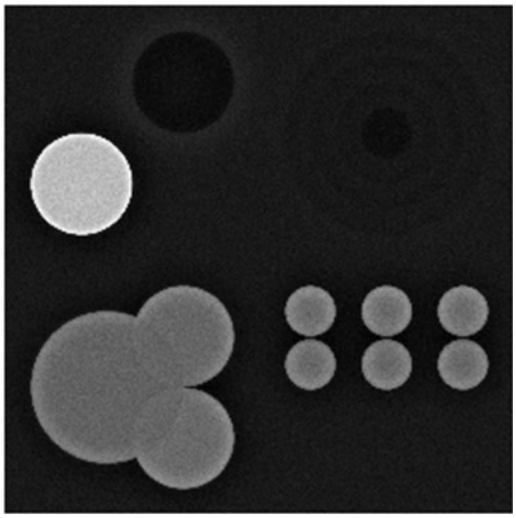


Figure 5.3: Simulated backscattered electron image.

is considered as a function HS defined by

$$HS : \begin{cases} E \to \mathbb{R}^D \\ x \mapsto v_i \end{cases}$$

where D is the number of abundance maps and E is the image domain (support space of pixels). This multivariate image can be also seen as set of D grey-scale images. We note HS a multispectral image at a low resolution. Let $\widetilde{HS} \in \mathbb{R}^{N_1 \times N_2 \times D}$ be a interpolated multispectral image whose spatial dimensions are N_1, N_2, which in our case corresponds to BSE image dimensions. Let $R \in \mathbb{R}^{N_1 \times N_2}$ be just the BSE image. We denote by $\widehat{HS} \in \mathbb{R}^{N_1 \times N_2 \times D}$ the multispectral image of the EDS abundances enhanced with the BSE image information, where HS_k is the k-th abundance map of image HS.

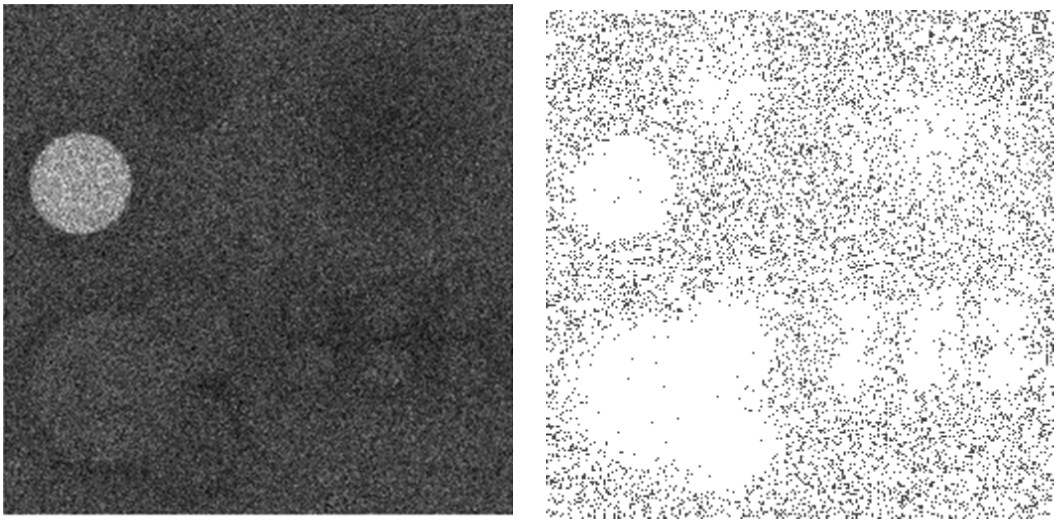
There are essentially three families of pansharpening techniques which are detailed as follows.

5.3.1 Component substitution methods (CS)

The purpose of information fusion techniques is to find the function ϕ satisfying

$$\widehat{HS} = \phi(\widetilde{HS}, R),$$

where $\widehat{HS}$ is "optimal" in a certain sense. The particular notion of optimality is precised below. Component substitution methods (CS) are based on the projection of the image HS into another space, separating the spatial information from the spectral one. The spatial information, i.e., contrast and contours between different objects, is often concentrated in a single grey-scale image. The main step of CS consists in replacing the image containing the spatial information by the spatially high resoluted image R. It is based on the assumption that the image containing only spatial information is highly correlated with R. Once the corrections are done, the data are projected back to the initial space. This approach is global and therefore



(a) (b)

Figure 5.4: (a) The pixelwise mean image of the simulated EDX image (256×256, $F = 1$). (b) Simulated Al Kα energy channel.

corrects all pixels with a single rule. The strength of this technique is its speed, however it depends on the fact that the image containing the spatial information of HS which must be similar in range and intensity distribution to R. If not, the merged image may have strong distortions. In our particular case, the BSE image has information which is not comparable with EDS images, that explains the artefacts which appears in our results, see Section 5.

The problem can be formulated mathematically as follows [91]:

$$\widehat{HS}_k = \widetilde{HS}_k + g_k(R - I_L), \tag{5.2}$$

where g_k corresponds to a corrective coefficient for the band k and I_L is the image containing all high resolution spatial information, i.e., typically

$$I_L = \sum_{i=1}^{D} w_i \widetilde{HS}_i, \tag{5.3}$$

where w_i is a coefficient depending on the degree of the spatial information of the spectral band i. Now let us present in a more detail way the most popular CS techniques.

CS by PCA.

Principal Component Analysis (PCA) [116, 69, 25] is a well-known dimensionality reduction technique, which aims at finding an orthonormal basis that maximizes the variance of the data. Thus, the data projected into this space summarize the statistically significant information. The fundamental assumption of this pansharpening technique is that the first component focuses all relevant spatial information while the other components contain secondary spatial and spectral contrast features. Using the CS model, the first principal component of $\widetilde{HS}$ is I_L, and w_i is the first column of the inverse transform. Note that this is not exactly the R which is used

in the Eq. (5.2), since the version of R should be histogram equalized with respect to I_L.

CS by Gram-Schmidt decomposition (GS).

The Gram-Schmidt (GS) technique is based on an orthogonal decomposition, which was invented by Kodak [75]. First, instead of considering only the cube $\widetilde{HS}$, one starts with the tensor composed of the concatenation of $\widetilde{HS}$ and R, thus of $D+1$ bands. Then, an orthogonal decomposition is performed on this new tensor. We find the vector of the basis that corresponds to the spatial information and it is replaced by R. Finally a reverse procedure is done to return to the initial representation space. The components of the Gram-Schmidt basis being orthogonal, the spatial information is expected to be orthogonal to all the other information sources. That is why the spatial information is, in theory, gathered in a single vector. In practice this is not always the case, especially if R contains dense information. Moreover this technique can be problematic in the case of multimodal SEM images since a significant orthogonal basis from EDX + BSE cannot be easily obtained without adding a kind of sparsity constraint.

5.3.2 Multiresolution analysis methods (MRA)

MRA techniques are founded on the application of a low pass filter to R, sometimes under different resolutions. Then, details are injected on $\widetilde{HS}$ thanks to the residue of R_L and R, which represents the high frequences of R. A mathematical formulation of this type of technique is written as [123, 153]:

$$\widehat{HS}_k = \widetilde{HS}_k + G_k \otimes (R - R_L), \tag{5.4}$$

where G_k corresponds to a corrective coefficient matrix for the band k and $\otimes$ is the multiplication term by term. Moreover, as mentioned above, $R - R_L$ represents the details that are injected in the low resolution image. Such family of techniques depends mainly on the type of decomposition performed for R_L and the diagonal matrix of gains G_k. Different kind of filters can be used for R_L: Gaussian filters, wavelets, mathematical morphology operators, etc. There are therefore many possibilities. Two techniques are now detailed.

MRA by Smoothing Filter-based Intensity Modulation (SFIM).

The technique [90] involves the use of a single low pass filter, noted H_{lp}, applied to R to get R_L. The enhancement is then obtained as:

$$\widehat{HS}_k = \widetilde{HS}_k + G_k \otimes (R - R * H_{lp}), \tag{5.5}$$

with

$$G_k = \widetilde{HS}_k \odot R * H_{lp}, \tag{5.6}$$

where $\odot$ is the division term by term. To achieve the low pass filter, a simple average filter is typically used. The formula can be simplified thanks to the coefficient matrix to obtain:

$$\widehat{HS}_k = R \otimes (\widetilde{HS}_k \odot R_l). \tag{5.7}$$

Thus it involves that the information of R was modulated by the ratio of the image $\widetilde{HS}_k$ and R_l. This allows integrating the contrast present in R without creating missing objects in the multispectral image $\widetilde{HS}$. Due to this pixel by pixel multiplication, there can be an issue on the low values of R_l, which requires to manage the dynamic of the image, such that it does not exceed a given range.

MRA by Laplacian Pyramid (MTF-GLP).

This technique [19, 97] has some similarities with the previous one. However, this time a multiple low-pass filter $H_{reso\ i}$ at various resolutions (i.e., at various scales) is used. More precisely, the low-pass filters are typically Gaussian convolutions. Missing information at a given scale i is injected into $\widetilde{HS}_{reso\ i+1}$ thanks to information from R at the resolution i. In our framework, the BSE image R and the multispectral abundance maps image of the EDS image are both in a multi-resolution structure pyramid, as illustrated in Figure 5.5. Details are injected at each resolution, until the nominal resolution of the BSE image is reached.

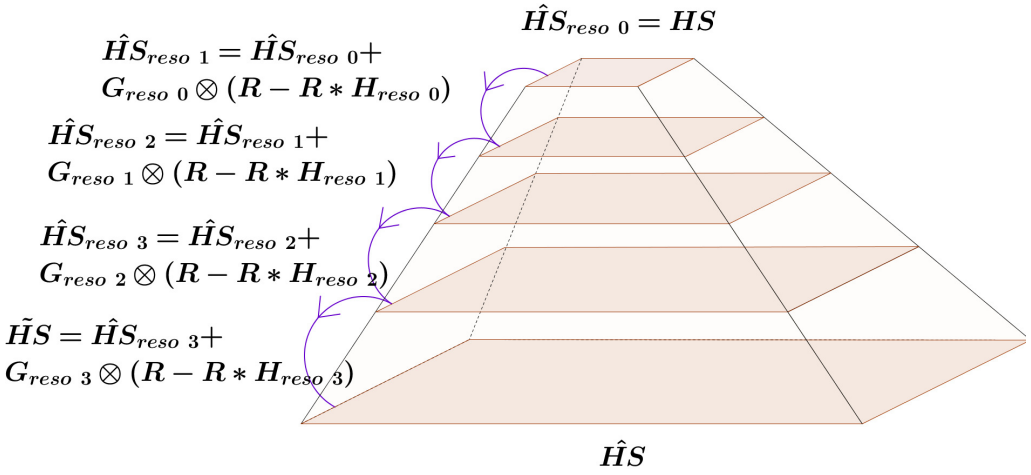


Figure 5.5: Model of a hierarchical decomposition of the information.

5.3.3 An hybrid method: Guided PCA

This technique [88, 73] consists in first doing a PCA on the multispectral image $\widetilde{HS}$. Then, on the d first principal components, a guided filter [64] is applied and nothing is done on the remaining $D - d$ bands. The rationale is based on the fact that the remaining bands correspond to "noise", and it would be useless to enhance the resolution of noise. Then a reversed PCA transform is made to obtain the original multispectral representation. The high resolution information from R is included by the guided filter. For more details about this technique, see [88, 73].

5.4 Fusion of SEM information by Abundance Guided Bilateral Filter (AGB)

As we have discussed above, when a pansharpening technique is used in the image fusion context, the starting point is a low spatial resolution image HS, together

with a high resolution "panchromatic" image R. Usually, the first step consists in upsampling HS, thanks to a basic bi-cubic interpolation to obtain the image $\widetilde{HS}$. Then, on this image at the nominal scale, the image enhancement is done, where the information is corrected by means of a particular pansharpening algorithm. The main issues with the previous techniques are that the CS approaches do not take into account the differences between the various components of the multimodal image, that is, the differences between the abundance maps themselves and the differences with the panchromatic image. In our case, there is a few correlation between the different EDX abundance maps. Moreover the MRA approaches do not consider the spectral link between the different abundance maps, since there is a finite quantity of material for each pixel (i.e., the sum of abundances at a given pixel is equal to 1).

In order to address these drawbacks, we propose an interpolation technique called abundance guided bilateral filter (AGB) by considering the relationships between the abundance maps. Thus, weights used in the interpolation would depend on both R and HS. Our approach of interpolation uses bilateral filter [151] and more exactly its cross version [42]. By the way, the bilateral filter has already been used on other super-resolution algorithms. Inspired by these works, we have chosen to improve the interpolation process which is a scaling process.

A bilateral filter is a nonlinear, edge-preserving denoising/regularizing operator for images. The intensity value at each pixel in an image is replaced by a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance between pixels on the grid, but also on their intensity (or more general radiometric) differences. Thanks to the fact that it uses spatial and range information, it preserves the edges, this is the reason why bilateral filter is used in super-resolution.

Formally, the joint bilateral filter of an image I guided by an image F is defined as :

$$I^*(x) = \frac{1}{W_p(x)} \sum_{x_i \in E} I(x_i) k(x, x_i), \tag{5.8}$$

where the kernel weights are given by

$$k(x, x_i) = f_r(\|F(x_i) - F(x)\|) g_s(\|x_i - x\|),$$

and where the normalization term is just given as:

$$W_p(x) = \sum_{x_i \in E} k(x, x_i),$$

such that f_r and g_s represent respectively the range (or spectral) and spatial kernels of parameters r and s. To simplify, we have chosen a Gaussian function for both kernels. However in our case we would like to consider the link between the different abundance maps. To handle this relationship, we need to define a guided function as a global criterion. Hence, we used the level of mixing of pixels, which will be represented by an order map: an image of ordered levels of intensity. In a way the order map has a low value if the pixel contains a mixture of various different materials and a high value if it is almost pure, so it contains contribution of few materials.

Let us precise how this order map is computed using the possible alternatives. The EDS spectrum at a position i of the image is a vector v_i which can be written as:

$$v_i = \sum_{r=1}^{D} a_{r,i} m_r + n_i, \tag{5.9}$$

where $\{m_r\}_{r=1}^{D}$ represent the set of D endmembers (spectral signature of the material), $a_{r,i}$ the abundance at vector i of each endmember r, and n_i an additive noise. This last term can be neglected. The nonnegative coefficients a_r we consider are convex combination such that $\sum_{r=1}^{D} a_r = 1$. By using this additional constraint, it is guaranteed to work on a $(D-1)$-simplex.

We have proposed in [50] different techniques to calculate such order map on abundance map images. We adopt here an approach based on the notion of majorization [114], which is a technique for ordering vectors of same sum. Let us consider two vectors $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n$ and $\mathbf{d} = (d_1, \ldots, d_n) \in \mathbb{R}^n$, then we say that C weakly majorizes D, written $\mathbf{c} \succ_w \mathbf{d}$, if and only if

$$\begin{cases} \sum_{i=1}^{k} c_i^{\downarrow} \geq \sum_{i=1}^{k} d_i^{\downarrow} \ , \forall k \in [1, n] \\ \sum_{i=1}^{n} c_i = \sum_{i=1}^{n} d_i \end{cases} \tag{5.10}$$

where $c_i^{\downarrow}$ and $d_i^{\downarrow}$ represent respectively the coordinates of C and D sorted in descending order. Majorization is not a partial order, since $\mathbf{c} \succ \mathbf{d}$ and $\mathbf{d} \succ \mathbf{c}$ do not imply $\mathbf{c} = \mathbf{d}$, it only implies that the components of each vector are equal, but not necessarily in the same order.

Let us define a majorization-like partial order adapted to the abundances. A permutation τ_i of the coordinates of the vectors v_i in the simplex is applied such that they are sorted in descending order. The majorization-like order $\leq_{\text{maj}}$ is defined as

$$v_i \leq_{\text{maj}} v_j \Leftrightarrow$$

$$\begin{cases} a_{\tau_i^{-1}(1),i} < a_{\tau_j^{-1}(1),j} \text{ or} \\ a_{\tau_i^{-1}(1),i} = a_{\tau_j^{-1}(1),j} \text{ and } a_{\tau_i^{-1}(2),i} < a_{\tau_j^{-1}(2),j} \text{ or} \\ \vdots \\ a_{\tau_i^{-1}(1),i} = a_{\tau_j^{-1}(1),j} \text{ and } \ldots \text{ and} \\ a_{\tau_i^{-1}(R),i} \leq a_{\tau_j^{-1}(R),j} \end{cases}$$

This order map between pixels brings a global information: the materials entropy. Since a pixel that has a high value with this order is less mixed than the other one. Let us write O the order map of the abundance images from the EDS image.

Figure 5.6 provides the corresponding O image for dataset 1 at the resolution of HS, together the BSE image R at the nominal resolution. Let us write $\widetilde{O}$ the order map image at the nominal resolution, where the value of all the missing pixels is put to zero.

We can now introduce the expression of the SEM fusion using the AGB filter:

$$\widehat{HS}(x) = \frac{1}{W_p(x)} \sum_{x_i \in E} HS(x_i) \widehat{k}(x, x_i) \tag{5.11}$$

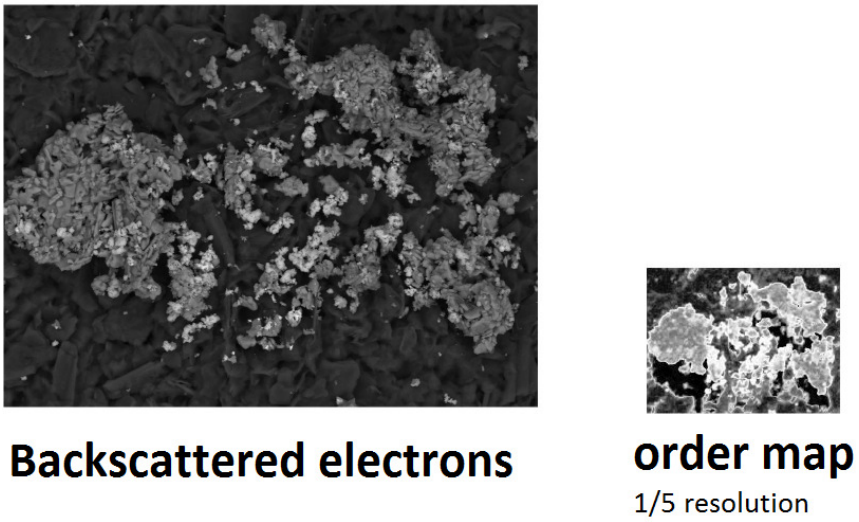


Figure 5.6: The two images used for the fusion of SEM information by AGB filter.

where the AGB kernel is written as the product of two kernels

$$\widehat{k}(x, x_i) = k_{space}(x, x_i) k_{EDX+BSE}(x, x_i) \tag{5.12}$$

given by

$$k_{space}(x, x_i) = g_s(\|x_i - x\|) M(\widetilde{O}(x_i)),$$

$$k_{EDX+BSE}(x, x_i) = g_s(|\widetilde{O}(x_i) - \widetilde{O}(x)|) g_s(|R(x_i) - R(x)|),$$

with a typical normalization term:

$$W_p(x) = \sum_{x_i \in E} \widehat{k}(x, x_i), \tag{5.13}$$

M is a mask that allows us to consider just the data on the values of abundance images available at the low resolution:

$$M(\widetilde{O}(x_i)) = \begin{cases} 1, & \text{if } \widetilde{O}(x_i) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{5.14}$$

and finally, g_s is a Gaussian function of scale parameter s.

5.5 Fusion of SEM information by Bilateral Guided Morphological Filter (BGM)

The goal now is to use mathematical morphology operators to build a nonlinear interpolation. This time, instead of using a classical average on the kernel, to approximate the missing data, we will use an alternative mean called the L^∞ barycenter, which can be seen related to mathematical morphology operators.

Let us first consider a grey-scale image $f : E \to \mathbf{T}$, $\mathbf{T} \subset \mathbb{R}$. The two basic operations in morphology are the grey-level erosion and the grey-level dilation whose definition respectively are:

$$\varepsilon_b(f)(x) = \inf_{h \in E}(f(x-h) - b(h)), \tag{5.15}$$

$$\delta_b(f)(x) = \sup_{h \in E}(f(x-h) + b(h)), \tag{5.16}$$

where b is a structuring function, which introduces the effect of the operators by the geometry of its support as well as its weights. Here we consider a structuring function with weights related to the bilateral kernel.

In order to compute the dilation and erosion, we obviously need a partial order for the supremum and infimum. Thus, we can use the order previously develop, represented on the image space by the order map O. Using this order, we can naturally define an erosion and dilation on the multivariate image of abundances as follows:

$$\varepsilon_b(HS)(x) = HS(x^a), \tag{5.17}$$

$$\text{where} \quad x^a = \arg\min_{h \in E}(O(x-h) - b(h)), \tag{5.18}$$

$$\delta_b(HS)(x) = HS(x^b), \tag{5.19}$$

$$\text{where} \quad x^b = \arg\max_{h \in E}(O(x-h) + b(h)). \tag{5.20}$$

Moreover, in our case, the dilation and erosion that we use involve the concept of changing the scale of the images. This notion is related is linked with the theory of morphological wavelets. We are not going to recall all the concepts that can be found in [66, 57], but at least to basic notions required for the rest.

In signal processing, the Shannon theorem states that the sampling of a signal, that is a discrete representation by a set of regularly sampled values of the signal, requires a frequency greater than Nyquist frequencies. In the case that this theorem is not respected the obtained signal may suffer from aliasing effects. This is the reason why Haralick et al. [62] defined a sampling condition for image processing that we are going to introduce.

Theorem 22 *Let $S \in \mathbb{Z}^2$ be a subset representing the sampling set of the grey-scale image f defined on the domain $E \subset \mathbb{Z}^2$, and let $K \in \mathbb{Z}^2$ be a flat structuring element (SE), i.e., a set defined at the origin. This structuring element is large enough to verify the following sampling condition if:*

$$\delta_K(S)(x) = E. \tag{5.21}$$

This condition is important to reconstruct the signal since it means that the structuring element is large enough to cover all E. According to [68] the sampling condition implies that the sampling distance must be less than half the distance of the structuring element K, that can be rewritten mathematically as:

$$\begin{cases} x \in K_y \Longrightarrow K_x \cap K_y \cap S \neq \emptyset \\ K \cap S = \{0, 0\} \end{cases} \tag{5.22}$$

where $K_x = \{k + x | k \in K\}$. Based on this assumption, Heijmans and Toet defined another general sampling strategy that we are going to review. Let E and S be two subsets $\mathbb{Z}^2$, E representing the support space of f and S the one of the sampled version of f. Let $\mathcal{K} : S \longrightarrow \mathcal{P}(E)$ be a mapping where $\mathcal{P}(E)$ represents the power space of all the subsets of E, where $\forall s \in S$, $\mathcal{K}(s) = \{k + s | k \in K\}$ that means that it is the translate of K along s. Then it is also possible to define the dual mapping as:

$$\begin{array}{rl} \mathcal{K}^* : & E \longrightarrow \mathcal{P}(S) \\ & \mathcal{K}^* = \{s \in S | x \in \mathcal{K}(s)\} \end{array} \tag{5.23}$$

Thanks to this definition, it follows that:

$$\mathcal{K}^{**} = \mathcal{K}$$
$$\cup_{s \in S} \mathcal{K}(s) = E \text{ if and only if } \forall x \in E,\ \mathcal{K}^*(x) \neq \emptyset.$$

That means that $\mathcal{K}$ covers E. Let us consider $f : E \to F$ a grey-scale image and $f_S : S \to F_S$ its sampling, and let us consider $\text{fun}(F)$ the set of grey-scale images and $\text{fun}(F_S)$ the set of grey-scale sampled images. It is now possible to define the dilation sampling operator:

$$\sigma_{\mathcal{K}}(f) : \left\{ \begin{array}{l} \text{fun}(F) \longrightarrow \text{fun}(F_S) \\ \sigma_{\mathcal{K}}(f)(s) = \sup\{f(x) | x \in \mathcal{K}(s)\} \end{array} \right. \tag{5.24}$$

and the erosion sampling operator:

$$\nu_{\mathcal{K}}(f) : \left\{ \begin{array}{l} \text{fun}(F) \longrightarrow \text{fun}(F_S) \\ \nu_{\mathcal{K}}(f)(s) = \inf\{f(x) | x \in \mathcal{K}(s)\} \end{array} \right. \tag{5.25}$$

The adjoint erosion of the dilation sampling operator is given by

$$\dot{\sigma}_{\mathcal{K}}(f_S) : \left\{ \begin{array}{l} \text{fun}(F_S) \longrightarrow \text{fun}(F) \\ \dot{\sigma}_{\mathcal{K}}(f_S)(s) = \inf\{f_S(s) | s \in \mathcal{K}^*(x)\} \end{array} \right. \tag{5.26}$$

and the adjoint dilation of the erosion sampling operator as

$$\dot{\nu}_{\mathcal{K}}(f_S) : \left\{ \begin{array}{l} \text{fun}(F_S) \longrightarrow \text{fun}(F) \\ \dot{\nu}_{\mathcal{K}}(f_S)(s) = \sup\{f_S(s) | s \in \mathcal{K}^*(x)\} \end{array} \right. \tag{5.27}$$

We can prove that σ and $\dot{\sigma}$ are adjoint operators using the standard procedure

$$\begin{aligned} \sigma_{\mathcal{K}}(f) \leq g_S \Leftrightarrow\ & \forall s \in S,\ \sup\{f(x) | x \in \mathcal{K}(s)\} \leq g_S(s) \\ \Leftrightarrow\ & \forall s \in S,\ \forall x \in \mathcal{K}(s),\ f(x) \leq g_S(s) \\ \Leftrightarrow\ & \forall x \in E,\ \forall s \in \mathcal{K}^*(x),\ f(x) \leq g_S(s) \\ \Leftrightarrow\ & \forall x \in E,\ f(x) \leq \inf\{g_S(s) | s \in \mathcal{K}^*(x)\} \Leftrightarrow\ f \leq \dot{\sigma}_{\mathcal{K}}(g_S) \end{aligned}$$

Identically we have that ν and $\dot{\nu}$ are adjoint too. Therefore, according to the fundamental theorem of mathematical morphology [67] , it is possible to build new operators by composition of the adjoint operators:

$$\rho_{\mathcal{K}} = \dot{\sigma_{\mathcal{K}}} \sigma_{\mathcal{K}} \tag{5.28}$$
$$\eta_{\mathcal{K}} = \dot{\nu_{\mathcal{K}}} \nu_{\mathcal{K}} \tag{5.29}$$

where ρ is a closing and η is an opening. Heijmans and Toet named σ and ν the sampling operators while ρ and η are called the reconstruction operators. We follow their terminology.

These sampling/reconstruction operators are at the basis of the morphological wavelet theory. We notice that these operators used a fixed structuring element K depending on just one image. By doing it, they do not use the intrinsic property of the images and so the sampling is not image-adaptive, moreover they cannot use the information from a second image. One would like that the sampling depends on both the original image and another image to have a morphological wavelet operators useful for image fusion.

Let us write $\tilde{\tilde{O}}^a$ the order map image at the nominal resolution (i.e., the resolution of image R), where the value to all the missing pixels is put to $\perp$. The order map image $\tilde{\tilde{O}}^b$ is that where the missing values are put to $\top$, where $\perp$ is smaller than all the values of O and $\top$ is higher than all the value of O. We introduce now two new operators:

$$\begin{aligned} \dot{\sigma}_{\mathcal{K}}(HS)(x) &= HS(x^b), && (5.30)\\ \text{where } x^a &= \arg\min_{h\in E}(\tilde{\tilde{O}}^a(x-h) - \mathcal{K}_x(h)), \\ \dot{\nu}_{\mathcal{K}}(HS)(x) &= HS(x^b), && (5.31)\\ \text{where } x^b &= \arg\max_{h\in E}(\tilde{\tilde{O}}^b(x-h) + \mathcal{K}_x(h)), \end{aligned}$$

This definition implies that $\dot{\sigma}_{\mathcal{K}}(HS)$ and $\dot{\nu}_{\mathcal{K}}(HS)$ are defined at the resolution of R.

In order to improve the accuracy of our morphological operators we choose to use bilateral structuring function, inspired by the theory developed in [76, 6]. The bilateral structuring function depends on x and its neighbourhood. However since we want to add the information from R, it should be also depending on the latter:

$$\mathcal{K}_x(h) = \begin{cases} \frac{1}{W_p} g_s(\|\tilde{\tilde{O}}^a(h) - \tilde{\tilde{O}}^a(x)\|) g_s(\|h-x\|) g_s(\|R(h) - R(x)\|) M(\tilde{\tilde{O}}(h)) & \text{if } h \in \tilde{\Omega}_x \\ -\infty & \text{elsewhere} \end{cases} \quad (5.32)$$

where the normalization term is:

$$W_p = \max_{h\in\Omega_x} \left(g_s(\|\tilde{\tilde{O}}^a(h) - \tilde{\tilde{O}}^a(x)\|) g_s(\|h-x\|) g_s(\|R(h) - R(x)\|) M(\tilde{\tilde{O}}(h)) \right). \quad (5.33)$$

We wrote the structuring function for the case of the locally adaptive erosion; in order to compute the corresponding locally adaptive dilation, one just has to replace $\tilde{\tilde{O}}^a$ by $\tilde{\tilde{O}}^b$ on the two previous formulae.

By means of these operators, it is possible to increase the scale of an image based on its own information but also the information from R. Using only one of the operators, the fused image may have a distortion by skewing the values to the high or low values. In order to correct this effect, we use as an enhanced image the mean between the adaptive upsampling erosion and dilation, i.e.,

$$\hat{HS} = \frac{\dot{\sigma}_{\mathcal{K}}(HS) + \dot{\nu}_{\mathcal{K}}(HS)}{2}. \quad (5.34)$$

This operator is just a kind of locally adaptive L^∞ barycenter.

5.6 Results and discussion

5.6.1 Evaluation criteria of pansharpening algorithms

The criteria used to evaluate the quality of the merged information are those conventionally considered in the context of the pansharpening literature. More precisely, we dealt with the following criteria:

C1: **The spectral distortion** between the enhanced multispectral image and the real multispectral image at the nominal resolution should be as small as possible. Or, in other terms, we would like to find the same materials for each pixel as the original image at high resolution;

C2: **The spatial distortion** between the enhanced multispectral image and the real one should not be too high.

Many alternative metrics can be used to quantify these two criteria [91]. Let us formaly precise those that we have used. We write $\widehat{\widehat{HS}}$ the multispectral abundance EDS image at the same resolution that R, that has been provided by the sensor at high resolution. In a way it is the ground truth such that our enhanced images $\widehat{HS}$ should be compared to $\widehat{\widehat{HS}}$. Moreover, we denote by $\widehat{\widehat{\mathcal{M}}} \in M_{n,D}(\mathbb{R})$ and $\widehat{\mathcal{M}} \in M_{n,D}(\mathbb{R})$ the two matrices representing respectively $\widehat{\widehat{HS}}$ and $\widehat{HS}$, where n is the total number of pixels (i.e., $n = N_1 \times N_2$), and D the number of abundance maps. We write by $\widehat{\mathcal{M}}_{i,:}$, $\forall i \in [1, n]$, a spectra of $\widehat{HS}$, and $\widehat{\mathcal{M}}_{:,k} \forall k \in [1, D]$ a map of $\widehat{HS}$. We have now all the notation to introduce the four parameters.

Cross correlation (CC) is a measure that evaluate the spatial distortion defined as

$$CC(\widehat{HS}, \widehat{\widehat{HS}}) = \frac{1}{D} \sum_{k=1}^{D} CCS(\widehat{\mathcal{M}}_{:,k}, \widehat{\widehat{\mathcal{M}}}_{:,k}), \tag{5.35}$$

where

$$CCS(\widehat{\mathcal{M}}_{:,k}, \widehat{\widehat{\mathcal{M}}}_{:,k}) = \frac{(\sum_{i=1}^{n} \widehat{\mathcal{M}}_{i,k} - \mu_{\widehat{\mathcal{M}}_{:,k}})(\sum_{i=1}^{n} \widehat{\widehat{\mathcal{M}}}_{i,k} - \mu_{\widehat{\widehat{\mathcal{M}}}_{:,k}})}{\sqrt{\sum_{i=1}^{n} (\widehat{\mathcal{M}}_{i,k} - \mu_{\widehat{\mathcal{M}}_{:,k}})^2 \sum_{i=1}^{n} (\widehat{\widehat{\mathcal{M}}}_{i,k} - \mu_{\widehat{\widehat{\mathcal{M}}}_{:,k}})^2}},$$

with $\mu_{\widehat{\mathcal{M}}_{:,k}} = n^{-1} \sum_{i=1}^{n} \widehat{\mathcal{M}}_{i,k}$ being the empirical mean. The CC is optimal when it is close to 1.

Spectral Angle Mapper (SAM) is a measure that assesses the spectral distortion by computing

$$SAM(\widehat{HS}, \widehat{\widehat{HS}}) = \frac{1}{n} \sum_{k=1}^{n} \widetilde{SAM}(\widehat{\mathcal{M}}_{i,:}, \widehat{\widehat{\mathcal{M}}}_{i,:}), \tag{5.36}$$

with

$$\widetilde{SAM}(\widehat{\mathcal{M}}_{i,:}, \widehat{\widehat{\mathcal{M}}}_{i,:}) = \arccos\left(\frac{< \widehat{\mathcal{M}}_{i,:}, \widehat{\widehat{\mathcal{M}}}_{i,:} >}{\|\widehat{\mathcal{M}}_{i,:}\| \|\widehat{\widehat{\mathcal{M}}}_{i,:}\|} \right),$$

where $< \cdot, \cdot >$ is the Euclidean inner product of vectors associated to the norm L^2, and where $\| \cdot \|$ is the norm L^2 of vectors. The SAM is optimal when it is near to 0.

Root mean squared error (RMSE) measures the mean residual error of fusion and is defined as:

$$RMSE(\widehat{HS}, \widehat{\widehat{HS}}) = \frac{\|\widehat{\mathcal{M}} - \widehat{\widehat{\mathcal{M}}}\|_F}{n.D}, \tag{5.37}$$

where $\| \cdot \|_F$ is the Frobenius norm of a matrix A, i.e., $\|A\|_F = \sqrt{\text{trace}(AA^t)}$. The RMSE is optimal when it is near to zero.

Synthetic adimensional global error (ERGAS) offers an overall measure of the quality of an enhanced image. It is given by the expression:

$$ERGAS(\widehat{HS}, \widehat{\widehat{HS}}) = \sqrt[100d]{\sum_{k=1}^{D} \left(\frac{RMSE(\widehat{\mathcal{M}}_{:,k}, \widehat{\widehat{\mathcal{M}}}_{:,k})}{\mu_{\widehat{\mathcal{M}}_{:,k}}} \right)^2},$$

where d is the ratio between the linear resolution of the BSE image R and the abundances EDS image HS, i.e.,

$$d = \frac{R\text{-linear spatial resolution}}{HS\text{-linear spatial resolution}}.$$

The ERGAS is optimal when close to 0.

We note that C1 and C2 are not enough to assess in our case the quality of the enhancement. Indeed the BSE images give not just access to the high resolution spatial information of the physical objects, but it provides other kind of information that can in theory allow us to have better result than just the EDX image at high resolution, such as the topographic shape of the sample. Thus we need to introduce a new criterion measuring the amount of information which is injected in the merged image:

C3: **The injected information.** Since the information provided by the various modalities can be relatively different, we would like to inject the useful one to improve segmentation and characterisation of the EDX image.

In order to quantify this criterion, we proposed to use the cross correlation between the norm of gradient of the images. To calculate such norm of the image gradient, different techniques can be used. In our studies, we computed the morphological gradient [142]. More precisely, we have:

Cross correlation gradient (CC_g) is a measure that evaluate the spatial distortion defined as

$$\text{CC}_\text{g}(R, \widehat{\widehat{HS}}) = \frac{1}{D} \sum_{k=1}^{D} CCS(\mathcal{R}g, \widehat{\widehat{\mathcal{M}}}g_{:,k}),$$

where $\mathcal{R}g$ represents the image gradient of the BSE image converted into a vector and $\widehat{\widehat{\mathcal{M}}}g_{:,k}$ is the image gradient of the enhanced abundance map at the nominal scale converted also into a vector.

An optimal enhancement method would be a compromise between the criterion C3 and the C1 and C2.

5.6.2 Evaluation on dataset 1

The EDS abundance maps of dataset 1 are provided in Figure 5.7. The results of the enhanced abundances $\widetilde{HS}$, obtained by the different techniques are provided in Figures 5.8, 5.9, 5.10, and 5.11 where the abundances are visualized as RGB color images. Quantitative results for this case according to the five measures are presented in Table 5.1.

From these results, we note that CS techniques are good to inject R on $\widehat{HS}$, while MRA techniques are good to increase the resolution. In addition, AGB, GFPCA and MTF-GLP present both a good compromise between these criteria. The BGM technique seems to be good to inject the information expect in the pixels with a high gradient.

Techniques	**CC**	**SAM**	**RMSE**	**ERGAS**	**CC_g**
GS	0.783	22.6	27.7	16.8	0.69
PCA	0.771	29.3	32.0	16.3	0.65
SFIM	0.831	11.6	19.6	14.4	0.29
MTF GLP	**0.899**	10.9	14.3	9.6	**0.70**
GFPCA	0.840	12.2	15.2	12.2	0.55
AGB	**0.90**	**9.5**	**11.2**	**9.35**	0.59
BGM	**0.90**	10.7	59.6	15.8	0.12

Table 5.1: Comparison of pansharpening algorithms for enhanced EDS abundance images of dataset 1.

5.6.3 Evaluation on dataset 2

In the case of this multimodal SEM dataset, which has a higher level of noise, we try to inject the BSE image R, depicted in Figure 5.12, to increase the resolution of the EDS abundance maps provided in Figure 5.13, and also to denoise the images. The results for our ABG method are given in Figure 5.14.

For comparaison, Figures 5.15 and 5.16 provide a visualization of the results obtained using the other methods. Quantitative results are given in Table 5.2, which lead to similar conclusions as for dataset 1.

Techniques	**CC**	**SAM**	**RMSE**	**ERGAS**	**CC_g**
GS	0.23	16.3	1.6	20.3	0.66
PCA	0.17	17.4	1.6	20.6	**0.67**
SFIM	0.14	15.9	1.74	27.9	0.17
MTF GLP	**0.30**	15.8	**1.4**	**20.1**	0.61
GFPCA	0.26	15.9	1.5	20.4	0.31
AGB	**0.30**	**15.7**	**1.4**	**20.1**	0.45
BGM	0.27	15.9	3.6	14.8	0.15

Table 5.2: Comparison of pansharpening algorithms for enhanced EDS abundance images of dataset 2.

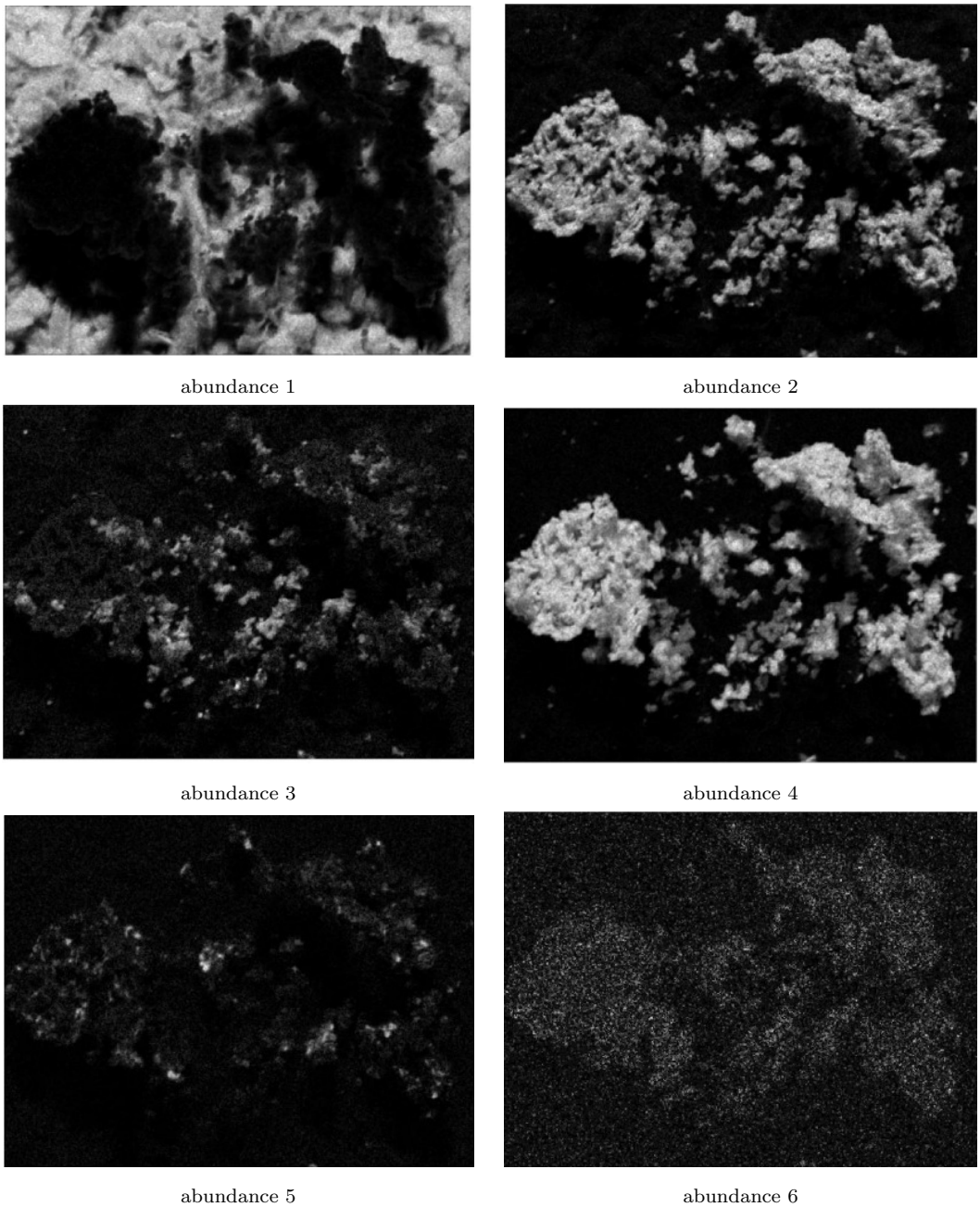


Figure 5.7: The six EDS abundance maps of SEM dataset 1 at the nominal resolution.

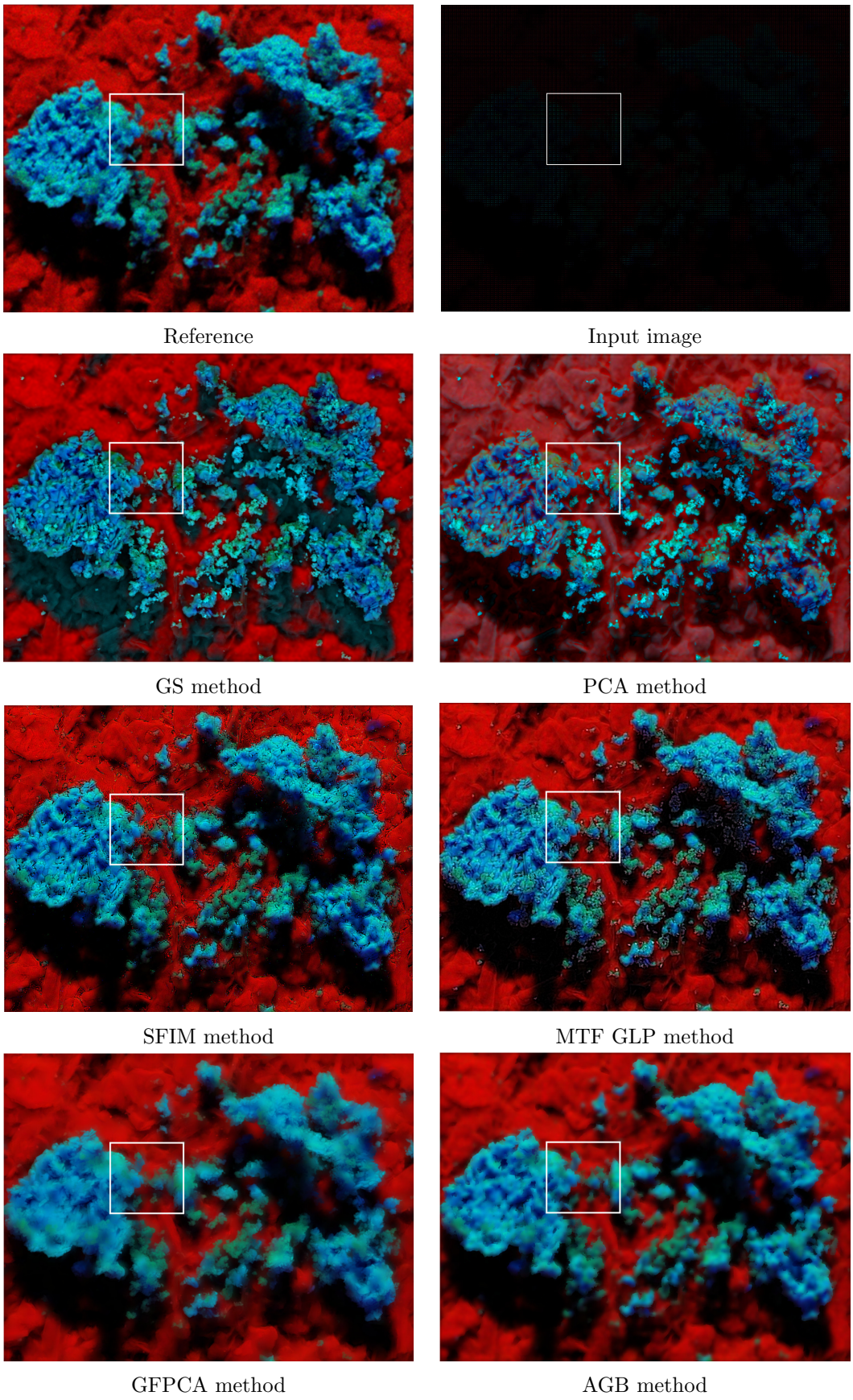
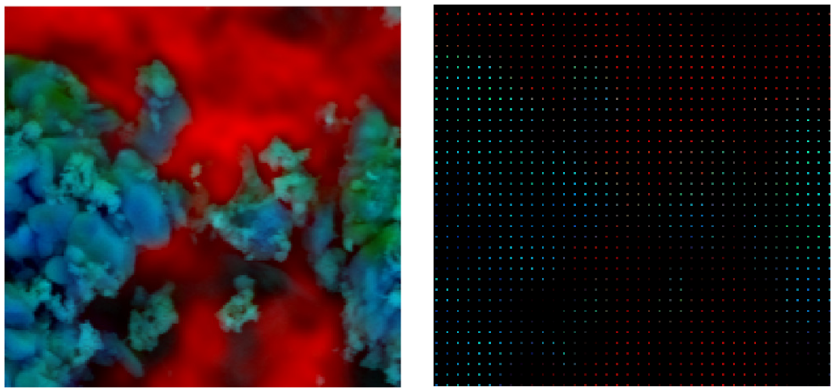
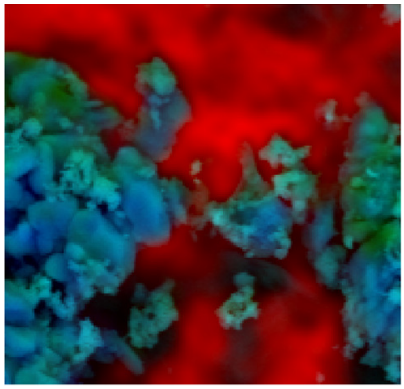


Figure 5.8: RGB color image from abundances 1,2,4 of the dataset 1 for the different pansharpening techniques.

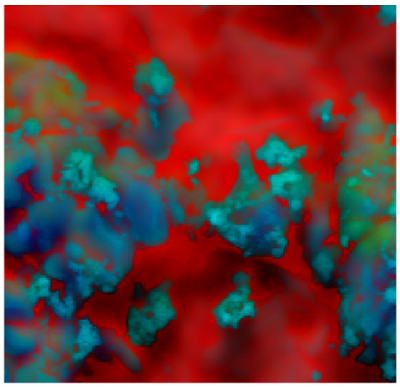


Reference

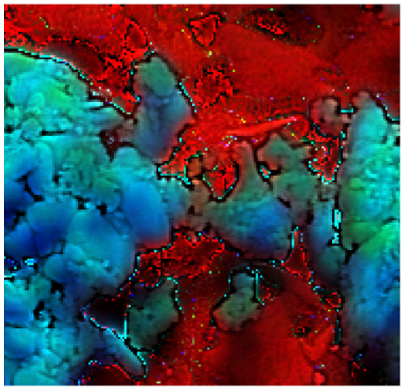
Input image



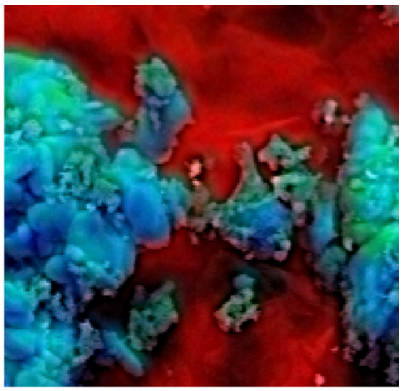
GS method



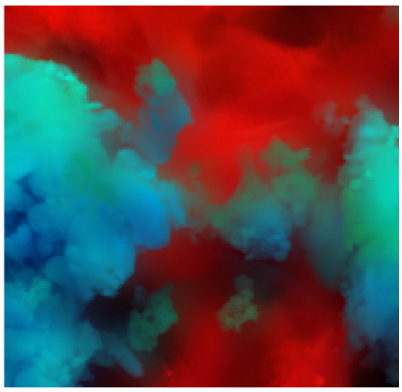
PCA method



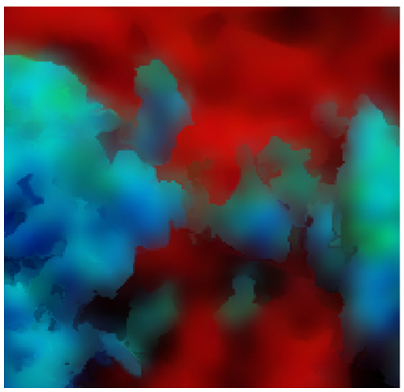
SFIM method



MTF GLP method



GFPCA method



AGB method

Figure 5.9: Zoom of the RGB color image from abundances 1,2,4 of the dataset 1 for the different pansharpening techniques.

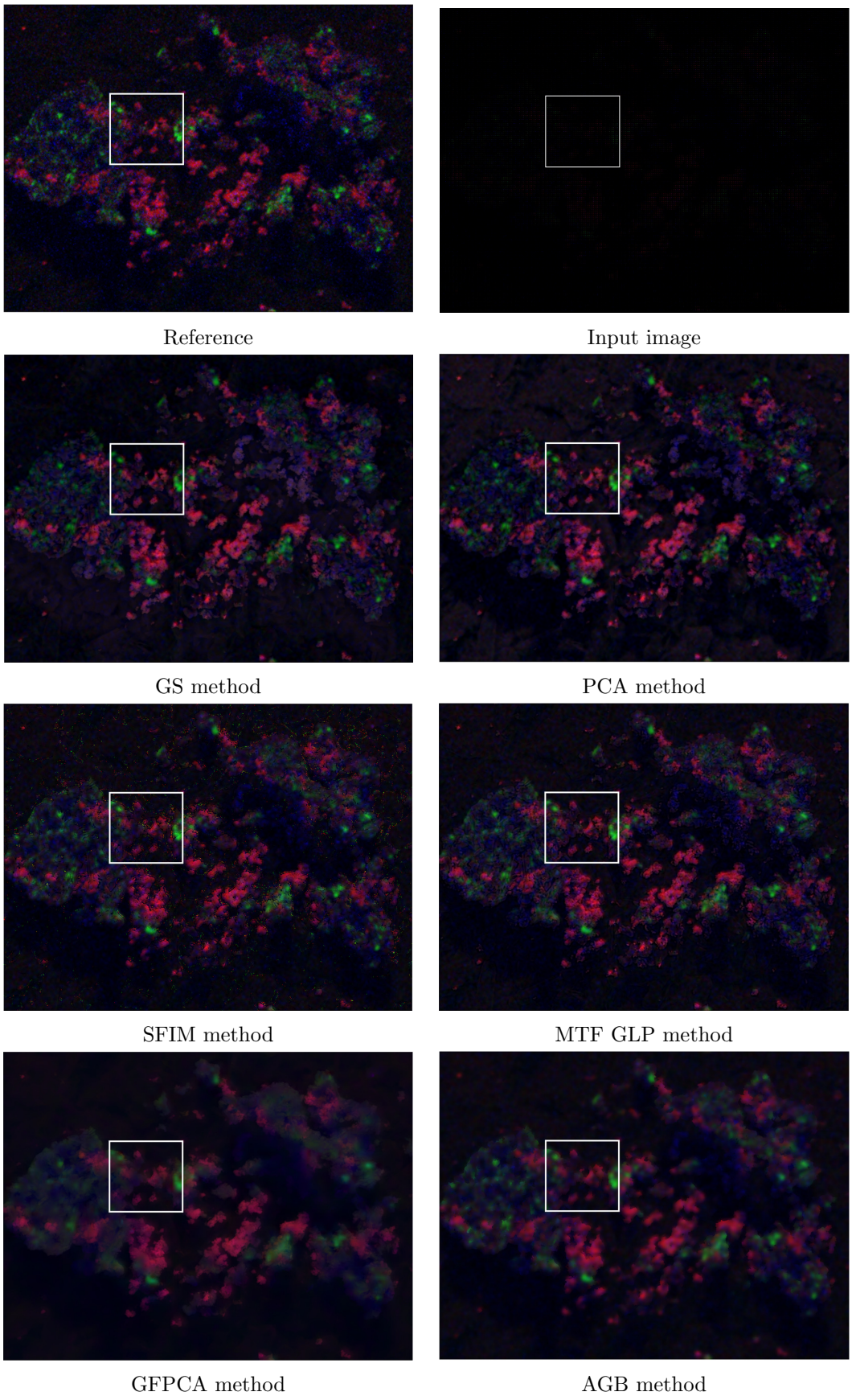
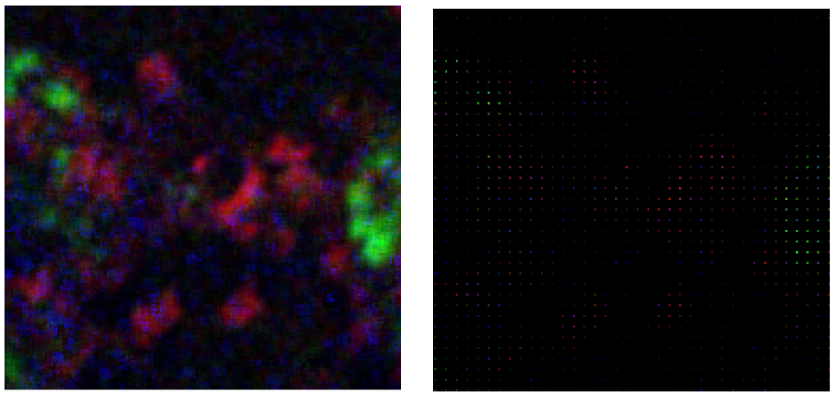
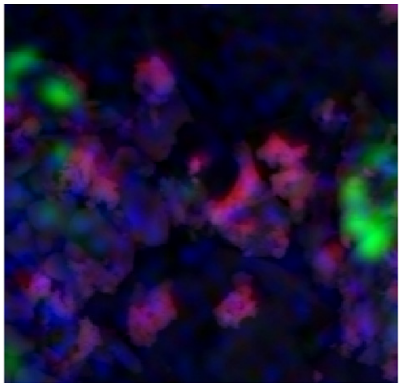


Figure 5.10: RGB color image from abundances 3,5,6 of the dataset 1 for the different pansharpening techniques.

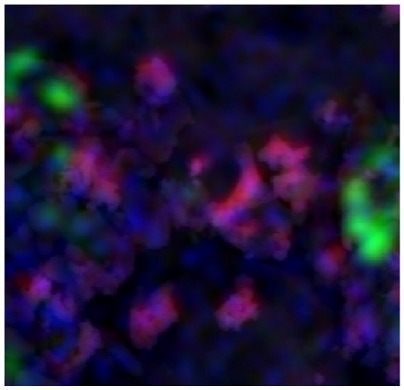


Reference

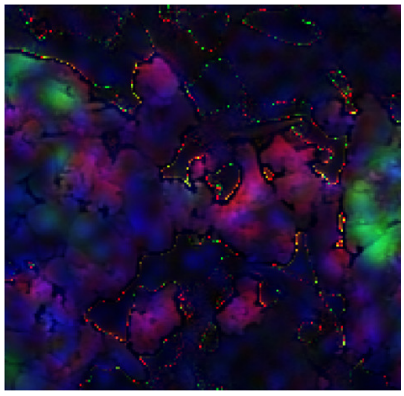
Input image



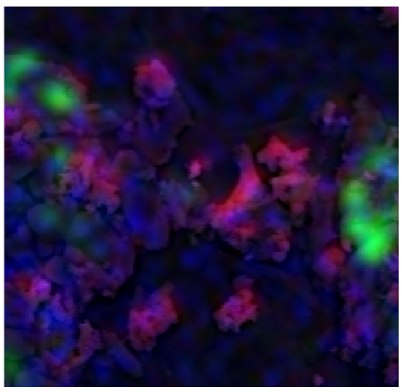
GS method



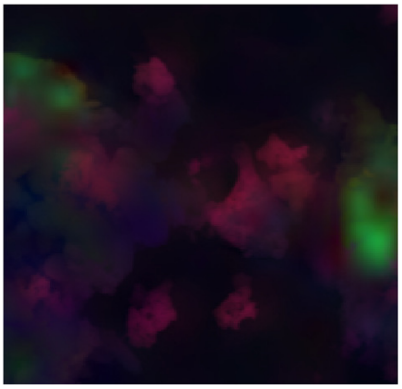
PCA method



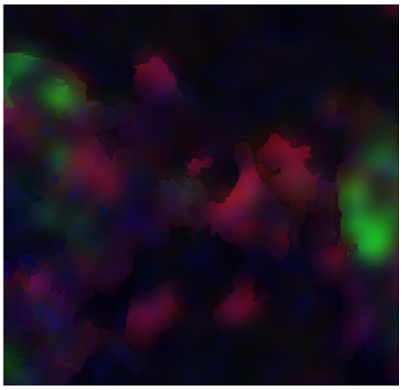
SFIM method



MTF GLP method



GFPCA method



AGB method

Figure 5.11: Zoom of the RGB color image from abundances 3,5,6 of the dataset 1 for the different pansharpening techniques.

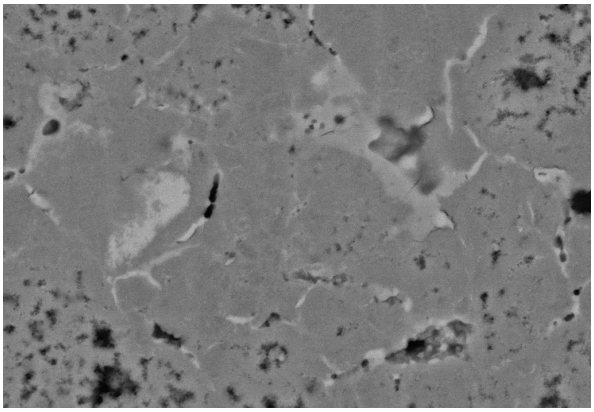


Figure 5.12: BSE image of SEM dataset 2.

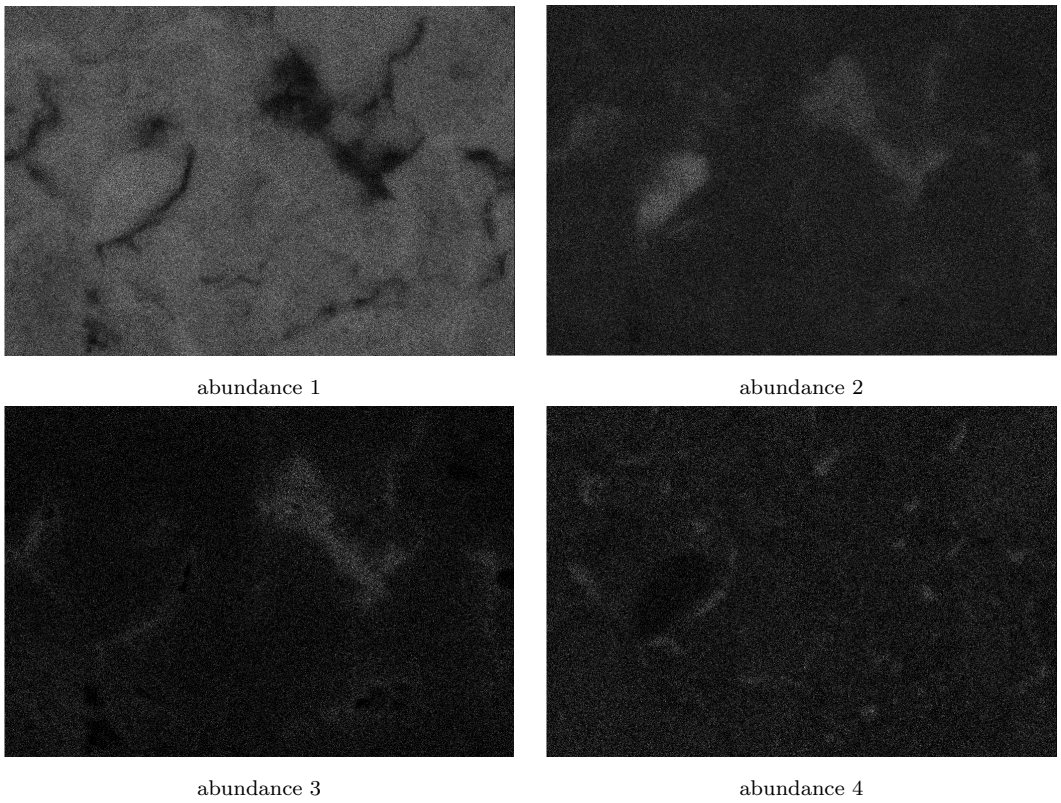


Figure 5.13: Four EDS abundance maps of SEM dataset 2 at the nominal resolution.

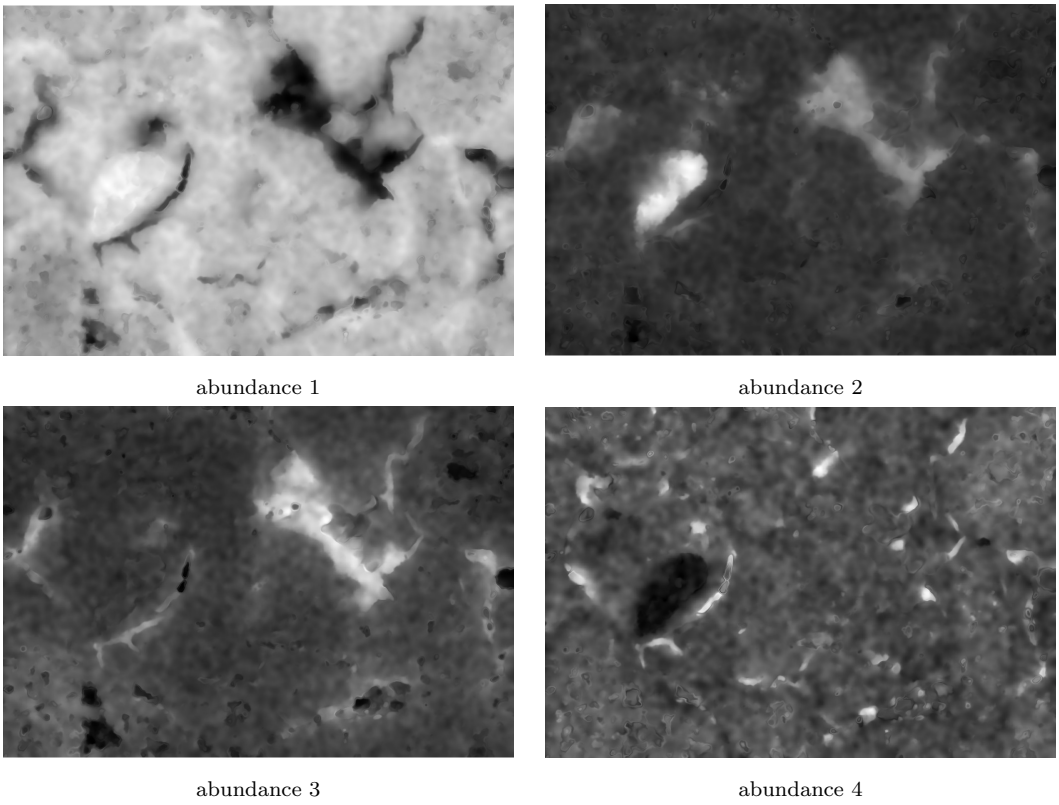


Figure 5.14: Four enhanced EDS abundance maps by means of ABG method of SEM dataset 2 at the nominal resolution.

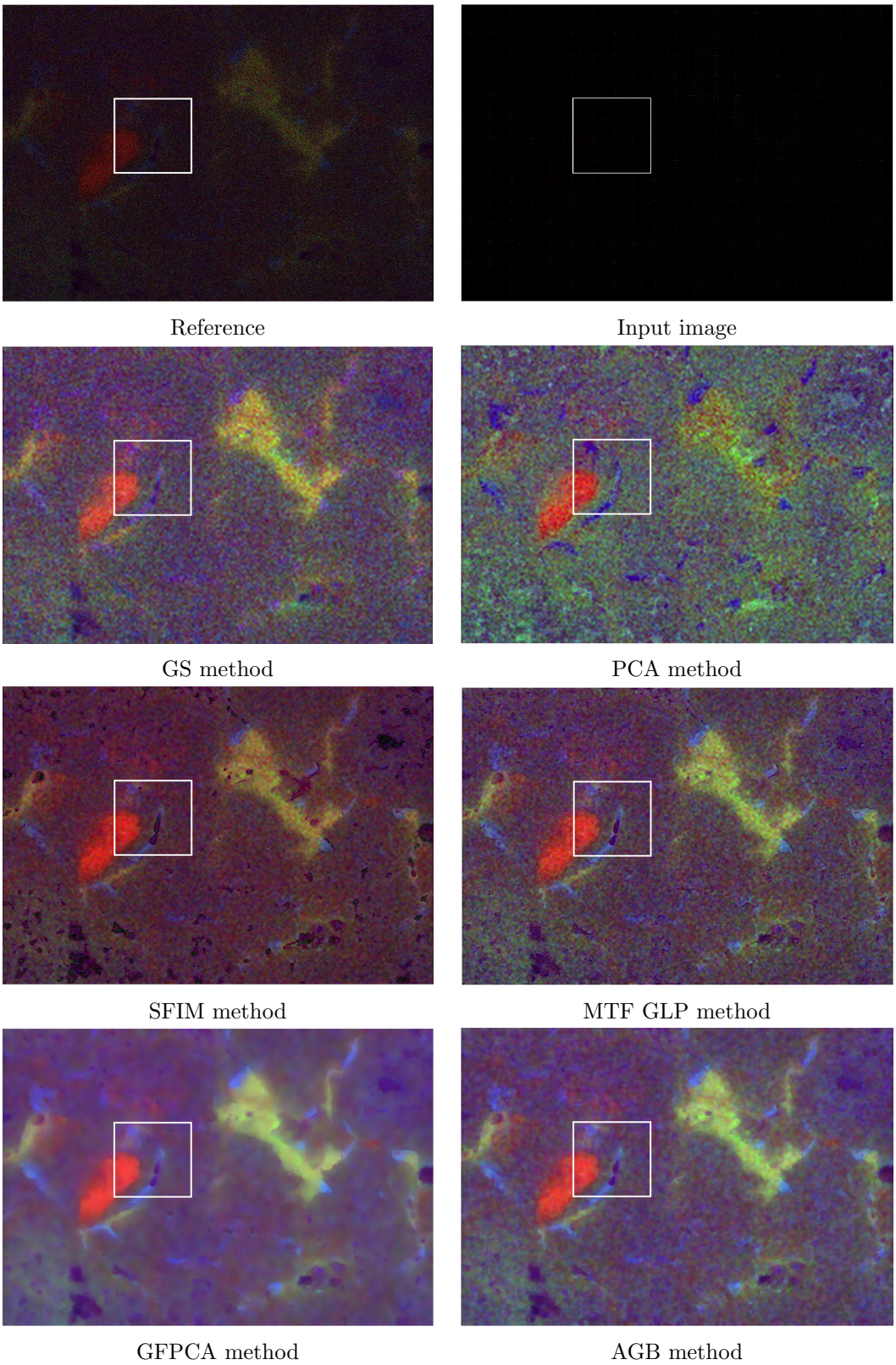
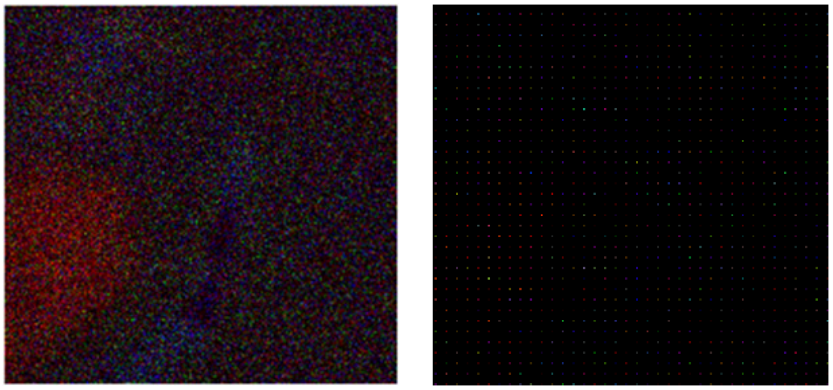
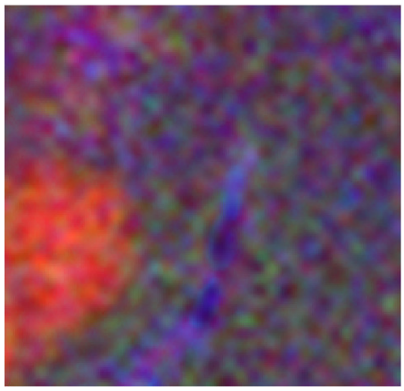


Figure 5.15: RGB color image from abundances 2,3,4 of the dataset 2 for the different pansharpening techniques.

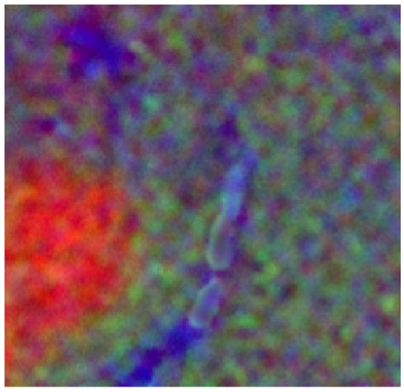


Reference

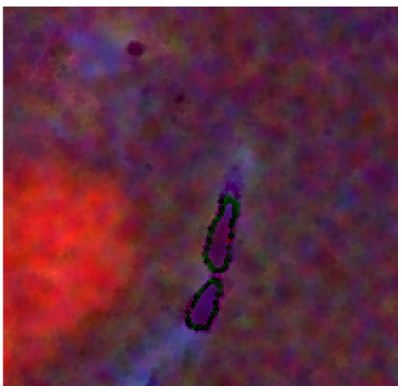
Input image



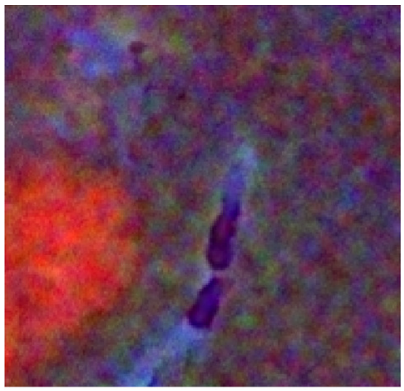
GS method



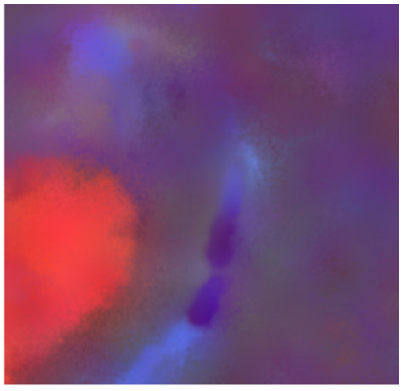
PCA method



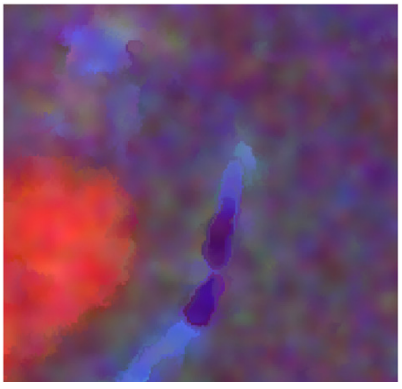
SFIM method



MTF GLP method



GFPCA method



AGB method

Figure 5.16: Zoom of the RGB color image from abundances 2,3,4 of the dataset 2 for the different pansharpening techniques.

5.6.4 Evaluation on simulated dataset

By means of the Monte Carlo software based on PENELOPE, we calculated various simulated datasets with different levels of noise and different resolutions. So first we have a dataset represented in Figure 5.2. It represents the sample at the full resolution without noise, thus it corresponds to the ground truth. The size of this image is $1024 \times 1024 \times 2048$. Then we simulate other EDX images of the same sample at resolution 4 times smaller than the original. The size of these images is therefore $256 \times 256 \times 2048$. First we extract the 7 abundance maps of these images and perform the fusion techniques. The results of the different algorithms are gathered in Table 5.3. Since we know the ground truth, for each pixel the most important materials that are present are known (see Figure 5.17). Hence after having performed the fusion techniques, for each pixel we extract the most significant material. This imply that, each pixel is represented by a number between 1 and 7. Then these images can be seen as the results of a classification. To evaluate the performance we used the overall accuracy (OA) which is defined as the percentage of pixels well classified on the whole image, and the average accuracy (AA) which represents mean of the accuracy of each class. These grey scale images are represented in Figure 5.17. One can see the results with the different levels of noise in Table 5.4. Our proposed ABG method and GFPCA give the best accuracies, even with high noise levels.

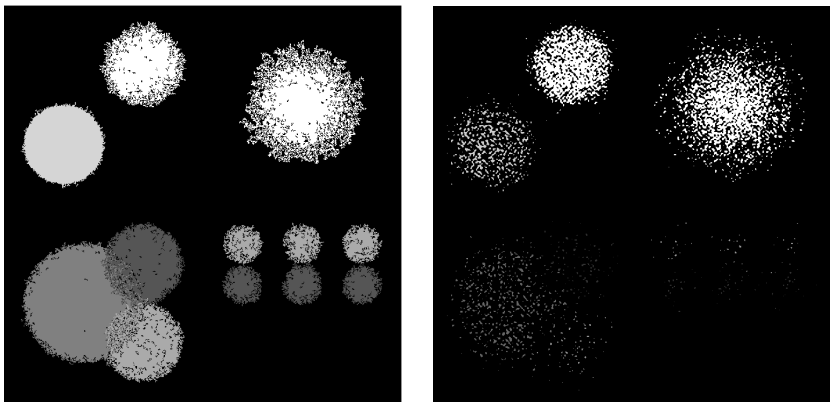
Techniques	**CC**	**SAM**	**RMSE**	**ERGAS**	**CC_g**
GS	0.065	26.72	0.51	445.61	0.12
PCA	0.046	26.52	0.62	461.6	0.20
SFIM	0.15	24.66	0.61	365.3	0.045
MTF GLP	0.19	24.0502	0.42	266.4	0.0763
GFPCA	0.29	**16.39**	**0.218**	439.4	0.1766
AGB	**0.36**	**16.30**	**0.217**	**420.6**	**0.3529**

Table 5.3: Comparison of pansharpening algorithms for enhanced EDS abundance images on simulated SEM image, forcing level 1.

5.7 Conclusion

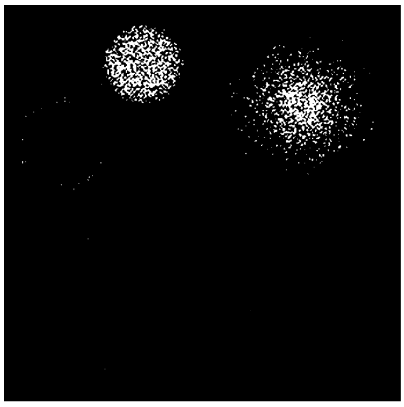
In this article, we have conducted a qualitative and quantitative evaluation of different algorithms of fusion of information applied on SEM images. Thanks to these algorithms we merged EDX and backscattered electrons SEM images. We compared global and local state of the art techniques and proposed two innovative one based on a guided bilateral filter, and a morphological approach. We also wrote a program based on the PENELOPE package to simulate large SEM-EDX spectral maps.

Thanks to these simulated and real SEM images, we evaluated the performances of the different techniques according to criteria defined for pansharpening problem. We also proposed one new criterion that is more adapted for the SEM segmentation problem. The accuracy of the pansharpening methods was assessed, and our proposed method was always found to be one of the best, together with the GFPCA. Our proposed algorithm performs well even on very noisy EDX images. Thanks to this fusion of information, we can increase the speed of the imaging process, since

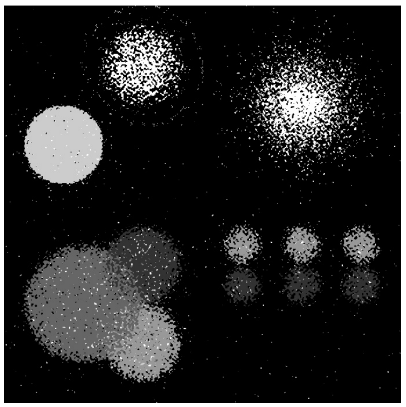


Reference

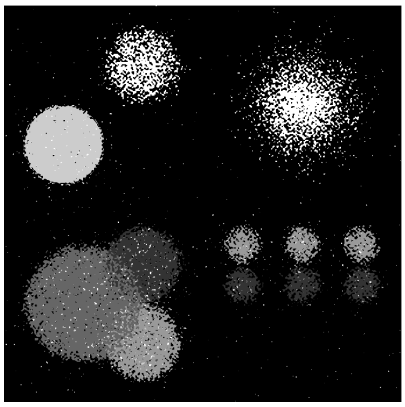
GS method



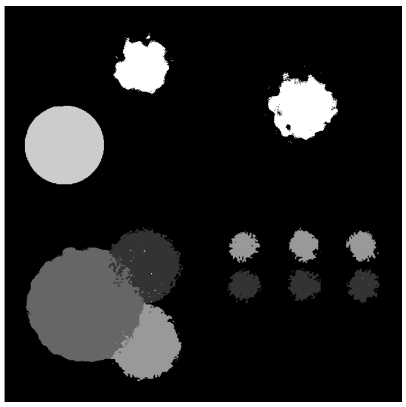
PCA method



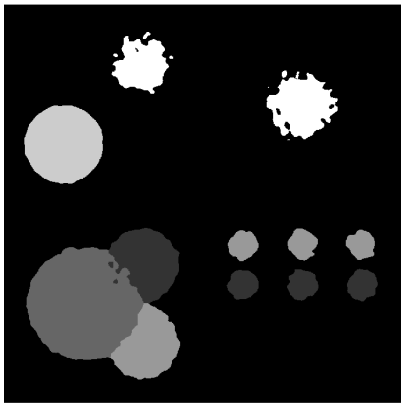
SFIM method



MTF GLP method



GFPCA method



AGB method

Figure 5.17: Extraction of the most abundant material at each pixel for the different pansharpening techniques.

Techniques	forcing level	OA	AA
GS	1	71.0	22.8
PCA	1	71.0	20.8
SFIM	1	70.0	33.3
MTF GLP	1	70.8	33.4
GFPCA	1	**78.2**	**36.9**
AGB	1	**78.9**	**36.9**
GS	10	74.9	23.0
PCA	10	75.3	21.5
SFIM	10	77.8	34.4
MTF GLP	10	77.9	34.4
GFPCA	10	**79.2**	**36.5**
AGB	10	**79.1**	**36.7**
GS	100	75.1	22.4
PCA	100	75.8	22.6
SFIM	100	78.3	35.3
MTF GLP	100	**79.1**	35.5
GFPCA	100	**79.2**	**36.5**
AGB	100	**79.2**	**36.8**

Table 5.4: Comparison of pansharpening algorithms for enhanced EDS abundance images on simulated SEM image, for different forcing levels.

we need fewer pixels. A way to improve the compression of this technique might be to consider pixels at given location, where the location is guided thanks to the backscattered image. By doing that, we would first take the backscattered image then have a look at the keypoints localizations, and then take the EDX image just at those points. Another way to improve this technique might be to consider the secondary electrons image. In addition, thanks to the developed formalism, it can easily be taken into account. In a broader perspective, the proposed guided bilateral filter could be used for other multispectral image pansharpening than SEM based ones.

Chapter 6

Spatial Regression for Image Pansharpening: Application to Multimodal SEM Image Fusion

Abstract

In this chapter, we compare ordinary kernel regression, kriging and Gaussian processes and apply such methods to multimodal multispectral SEM image fusion. The kernel regression that we use is the technique presented in the previous chapter, called Abundance Bilateral Guided (ABG). We compare it mathematically and experimentally to ordinary kriging. In addition we propose a way to perform image fusion of information by means of the ordinary kriging and show the interest of the approach for image pansharpening. By combining the different SEM modalities, we increase the resolution of the EDX multispectral image, and provide more details on the physicochemical composition of samples obtained from the enhanced EDX.

Résumé

Le microscope électronique à balayage (MEB) permet d'acquérir des images à partir d'un échantillon donné en utilisant différentes modalités. Le but de ce chapitre est d'analyser l'intérêt de la fusion de l'information pour améliorer les images acquises par MEB. Nous avons mis en oeuvre différentes techniques de fusion de l'information des images, basées dans ce chapitre sur la théorie de la régression spatiale. Ces solutions ont été testées sur quelques jeux de données réelles.

6.1 Introduction

In this chapter, we propose an original approach of spatial fusion of information able to increase the resolution of multispectral images.

This technique is called pan sharpening. Pan sharpening aim at merging different spatial and spectral information. A huge variety of techniques exists in remote sensing [75], however in Scanning Electron Microscopy these kind of techniques are

not so used. We express in this chapter this problem as a problem of spatial regression. To perform this regression we consider different techniques. We compare them mathematically and propose one innovative technique based on regularised kriging. We note that a solution based on Parzen windows was introduced in the previous chapter. So we present the mathematical aspect of this solution, and compare it to our solution based on kriging. Kriging [102, 28, 33, 101, 32] is an interpolation technique used originally in geosciences for the evaluation of minerals deposit spatial repartition. This method was developed by Georges Matheron departing from some studies from Daniel G. Krige. The technique based on kriging is inspired by the innovative work [34] where the author first performed a principal component analysis followed by kriging applied on each principal component. By combining this information with other modalities they were able to recognize if a patient was infected by the malaria or not. We note also the innovative work proposed by [84], where the authors use local gaussian process regression techniques to increase the resolution of an image. Given an image the authors randomly extract patches from a blurred version of the image and on its corresponding high-resolution version. Then given pairs of patches they learn thanks to a technique based on multiple Gaussian process regressions how to transform a low resolution patch into a high resolution one. Even if this technique is quite interesting, the Gaussian process regression they used does not take into consideration the full spatial information of images.

That is why, here we propose to use a local ordinary kriging where the coefficients depend on the information of the other modalities we want to inject. Then we are able to reconstruct the missing information while injecting other modalities information. To present our work we first introduce the kernel regression, then gaussian process regression, then ordinary kriging. We provide results of these techniques on the samples presented in the previous chapter.

6.1.1 Notations

We consider that an image f of dimension D with support space of pixels $E \subset \mathbb{Z}^2$ can be defined as a function:

$$f := \begin{cases} E \to F \subset \mathbb{R}^D \\ x \mapsto f(x) \end{cases} \tag{6.1}$$

In the sequel, we limit ourselves to two kinds of images from multimodal SEM. We first consider the backscattered electron (BSE) image, which is denoted R. It is a gray scale image, i.e., $D = 1$, of spatial dimensions $N_1 \times N_2$. We also consider a multispectral image HS which is in our case the result of the abundance maps, at a lower resolution, of dimension $n_1 \times n_2 \times D$, where D is the number of abundance maps. We additionally denote $\widehat{HS}$ the abundance map multispectral image at the nominal spatial resolution: size $N_1 \times N_2 \times D$. The goal of image fusion is just to increase the spatial resolution from $n_1 \times n_2$ to $N_1 \times N_2$.

We consider that an image g is the realization of a random function G. Let us consider a probability space (Ω, A, P) and a domain $\mathcal{D} \in E$. A random field, or random function, on the spatial domain $\mathcal{D}$ with values in F is a function of two variables, denoted $Z(x, \omega)$. For each $x \in D$, $G(x, :) : \omega \to G(x, \omega)$ is a random variable on (Ω, A, P). Moreover, for each $\omega_0 \in \Omega$, $G(: \omega_0) : x \to G(x, \omega_0)$ is a

function of $D \rightarrow E$. We write $G(x)$ the random function at the position x, and $g(x)$ the regionalized function.

We note that in our case we deal with image data hs that are abundance map. That means that at a given position x, we have $\sum_{i=1}^{D} hs_i(x) = 1$, together with $hs_i(x) \geq 0, \forall x \in E, \forall i$. This means that the data are on a convex set of the positive orthant of $\mathbb{R}^D$, called $D-1$ simplex. From a physical viewpoint, $hs_i(x)$ represents a quantity (between 0 and 1) of material i at pixel x.

6.2 Spatial kernel regression

Let us consider that the image hs is composed of N spatial positions $\mathcal{N}_1 = \{x_i | i \in [1, N]\}$, such that we have the value of hs in just n spatial positions $\mathcal{N}_2 = \{x_i | i \in [1, n]\}$. Our goal is to estimate the value of hs in the remaining $N - n$ spatial positions $\mathcal{N}_3 = \{x_i | i \in [1, N]/[1, n]\}$. Or, more formally, given a vector of inputs $\mathcal{N}_2 = (x_1, x_2, ..., x_n)^t$, we will predict the output hs for new positions x^*. The estimated predictor is denoted $\widehat{hs}(x^*)$.

Let us consider that the available information that we have on each observation is most of the time corrupted by Gaussian noise of zeros mean and standard deviation σ. Hence, we can write the model in the form:

$$\widehat{hs}(x_i) = f(x_i) + \mathcal{N}(O, \sigma) \tag{6.2}$$

where f is the true information. By considering that the different data points $(x_i, hs(x_i))$ are independent and identically distributed (i.i.d.), we can evaluate the likelihood as follows:

$$\mathcal{P}(HS = hs(\mathcal{N}_2) | \mathcal{N}_2, f, \sigma) = \prod_{i=1}^{n} \mathcal{P}(HS = hs(x_i) | x_i, f, \sigma), \tag{6.3}$$

with $hs(\mathcal{N}_2) = (hs(x_1), \ldots, hs(x_n))^t$.

$$\mathcal{P}(HS = hs(\mathcal{N}_2) | X, f, \sigma) \;=\; \prod_{i=1}^{n} \mathcal{N}(hs(x_i) | x_i, f, \sigma) \tag{6.4}$$

$$\mathcal{P}(HS = hs(\mathcal{N}_2) | \mathcal{N}_2, f, \sigma)) \;=\; \frac{1}{\sqrt{2\pi}^n \sigma^n} . e^{\frac{1}{2\sigma^2} \| f(\mathcal{N}_2) - hs(\mathcal{N}_2) \|^2} \tag{6.5}$$

We seek a function $f(x)$ for predicting $hs(x)$ given values of the input $\mathcal{N}_2$. However, the optimal function should not depend on $\mathcal{N}_2$. This leads us to a criterion for choosing f, in the sense that the optimal function f should minimize:

$$\begin{aligned} \mathcal{R}(f) &= \mathbb{E}_{hs,x}(\| f(x) - hs \|^2) \\ &= \mathbb{E}_x \mathbb{E}_{hs|x}(\| f(x) - hs \|^2), \end{aligned} \tag{6.6}$$

which can be rewritten as:

$$f(x) = \arg\min \mathbb{E}_{HS|X=x}(\| f(x) - hs \|^2), \tag{6.7}$$

and its solution is just:

$$f(x) = \mathbb{E}_{HS|X=x}(hs), \tag{6.8}$$

which corresponds to the conditional expectation, also called the regression function. Then the best prediction needs $\mathcal{P}(HS|X = x)$. However, we do not have access to this probability and the goal here is to show how to evaluate it thanks to kernel density estimation also known as Parzen-Rosenblatt window method [126].

We consider a model of the data which is similar to the one introduced in [16]. Let us write $T = (HS, X)$ the joint random variable. For the sake of simplicity, we write $p(t)$ the corresponding joint distribution with $t \in \mathbb{R}^{D+2}$. Let us take a (small) region $\mathcal{R} \subset \mathbb{R}^{D+2}$ containing t. The probability of this region is just:

$$\mathcal{P} = \int_{\mathcal{R}} p(t)dt. \tag{6.9}$$

We suppose that out of the n points there are K points that fall into $\mathcal{R}$, such that the K points follow a binomial distribution, i.e.,

$$\text{Bin}(K|n, \mathcal{P}) = \frac{n!}{K!(n-K)!}\mathcal{P}^K(1-\mathcal{P})^{n-K}. \tag{6.10}$$

If we want the mean number of points falling in $\mathcal{R}$ to be equal to K, that means that $K = \mathcal{P}n$. Moreover, we can assume that $\mathcal{R}$ is small in the sens that $\mathcal{P} = V\,p(t)$ is a good approximation, where V is the volume of $\mathcal{R}$. Hence, under this assumption, one has

$$p(t) = \frac{K_t}{nV}, \tag{6.11}$$

where K_t represents the mean number of point falling into $\mathcal{R}$. Then, we can separate $\mathbb{R}^{D+2}$ into a discrete set of bins and count the number of points falling in each bin. Or we can use a Parzen window implanted at each point and integrate the number of points around each window, which is a smoother estimation. More formally, we have

$$K_{hs,x} = \sum_{i=1}^{n} k(t - t_i, \sigma_1).$$

In previous chapter, we used for ABG interpolation a Parzen window method with a Gaussian (or rbf) kernel: $k(t_i - t, \sigma_1) = \exp(\|t_i - t\|^2/(2.\sigma_1))$. But, there are several possible kernels that one can consider classicaly in density estimation,

- triangle kernel:

$$k(u) = (1 - |u|)\ 1_{(|u|\leq 1)}K(u) = (1 - |u|)\ 1_{(|u|\leq 1)}, \tag{6.12}$$

- Epanechnikov kernel:

$$k(u) = \frac{3}{4}(1 - u^2)\ 1_{(|u|\leq 1)}K(u) = \frac{3}{4}(1 - u^2)\ 1_{(|u|\leq 1)}, \tag{6.13}$$

- quadratic kernel:

$$k(u) = \frac{15}{16}(1 - u^2)^2\ 1_{(|u|\leq 1)}K(u) = \frac{15}{16}(1 - u^2)^2\ 1_{(|u|\leq 1)}, \tag{6.14}$$

- cubic kernel:

$$k(u) = \frac{35}{32}(1-u^2)^3\ 1_{(|u|\leq 1)}K(u) = \frac{35}{32}(1-u^2)^3\ 1_{(|u|\leq 1)}, \tag{6.15}$$

- Gaussian kernel:

$$k(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}K(u) = \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}u^2}, \tag{6.16}$$

- circular kernel:

$$k(u) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)1_{(|u|\leq 1)}K(u) = \frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)1_{(|u|\leq 1)}. \tag{6.17}$$

Let us continue using the Gaussian kernal, such that we can write

$$K_{hs,x} = \sum_{i=1}^{n} k(hs - hs_i, \sigma_1)k(x - x_i, \sigma_1).$$

The multiplication of the two Parzen windows helps us to bring both the spatial and spectral information. Hence the density function is now:

$$p(t) = p(hs, x) = \frac{1}{n}\sum_{i=1}^{n} k(x - x_i, \sigma_1)k(hs - hs(x_i), \sigma_1), \tag{6.18}$$

As we seek for $\mathcal{P}(HS = hs(x)|x, \omega)$, we use Bayes theorem to obtain:

$$\mathcal{P}(HS = hs|x, \omega) = \frac{n^{-1}\sum_{i=1}^{n} k(x - x_i, \sigma_1)k(hs - hs(x_i), \sigma_1)}{\int p(hs, x)dhs}. \tag{6.19}$$

Then by applying the expression (6.8), it is obtained:

$$\begin{aligned} f(x) &= \int hs\frac{\sum_{i=1}^{n} k(x - x_i, \sigma_1)k(hs - hs(x_i), \sigma_1)}{\int p(hs, x)dhs}dhs \\ &= \frac{\sum_{i=1}^{n} n^{-1}\int hs\, k(x - x_i, \sigma_1)k(hs - hs(x_i), \sigma_1)dhs}{\sum_{i=1}^{n}\int k(x - x_i, \sigma_1)k(hs - hs(x_i), \sigma_1)dhs}. \end{aligned} \tag{6.20}$$

Using the fact that $\int hs\, k(hs - hs(x_i), \sigma_1)dhs = hs(x_i)$, and changing the variable we finally have:

$$\widehat{hs}(x) = \frac{\sum_{i=1}^{n} hs(x_i)k(x - x_i, \sigma_1)}{\sum_{i=1}^{n} k(x - x_i, \sigma_1)}. \tag{6.21}$$

In this model, and going back to the problem of multimodal SEM, we do not consider the information brought by the other modalities that can be useful to improve the estimation of $\widehat{hs}(x)$. In order to deal with this idea, and as suggested by [138] in section 4.3 chapter 4, we modify the single kernel by a product of kernels:

$$k(x - x_i) = \prod_j k_j(x - x_i, \sigma_j).$$

In particular, the Parzen window kernel will be:

$$k(x - x_i, \sigma_1) = k(x - x_i, \sigma_1)k(r(x) - r(x_i), \sigma_1)k(\widetilde{hs}(x) - \widetilde{hs}(x_i), \sigma_1), \quad (6.22)$$

such that this kernel allows us to evaluate the coefficients of the window using the spatial and spectral information, as long as the information of the BSE image r. The image estimate will be given by:

$$\widehat{hs}(x_j) = \sum_{i=1}^{n} hs(x_i)\omega((x_j - x_i)\sigma_1^{-1}) \quad (6.23)$$

with the normalized weights

$$\omega((x_j - x_i)\sigma_1^{-1}) = \frac{k(x_j - x_i, \sigma_1)}{\sum_{i=1}^{n} k(x_j - x_i, \sigma_1)}.$$

This notation is correct with the rbf kernel.

The question that we can ask ourselves now is what are the statistical properties of such estimator? Indeed, we would like to be sure that if we use another "training set" $\mathcal{N}_2 = \{x_i | i \in [1, n]\}$, with $\{x_i | i \in [1, n]\} \overset{\text{i.i.d.}}{\sim} P(x)$, the regression would stable. Moreover, we do not have an unlimited access to data point and therefore the influence of the number of data should be understood. Finally, we need to know the influence of the (smoothing) scale parameter σ_1.

Let us denote the real function $hs(x)$ and let $\widehat{hs}(x, \mathcal{N}_2)$ be the function that is estimated using a Parzen window of parameter σ_1 from training set $\mathcal{N}_2$. We have the following statistical properties for the normalized kernel function:

$$\int w(u)du = 1, \quad \int uw(u)du = 0, \quad \int u^2 w(u)du = \sigma_k^2.$$

Let us first evaluate the bias of the estimator:

$$\text{bias}^2 = \int \left(\mathbb{E}_{\mathcal{N}_2}\left(\widehat{hs}(x, \mathcal{N}_2) - hs(x)\right)\right)^2 P(x)dx, \quad (6.24)$$

thus we have

$$\text{bias}^2 = \int \left(\mathbb{E}_{\mathcal{N}_2}\left(\sum_{i=1}^{n} hs(x_i)\omega((x_i - x)\sigma_1^{-1}) - hs(x)\right)\right)^2 P(x)dx. \quad (6.25)$$

We can consider that the function hs is locally smooth of first order. This hypothesis is justified by the fact that the function is smoothed by a local filter for denoising it. Under this assumption, a Taylor expansion can be compputed, i.e.,

$$hs(x_i) = hs(x) + (x - x_i)hs'(x) + \frac{(x - x_i)^2}{2}hs''(x) + O((x - x_i)^2) \quad (6.26)$$

Using this approximation in (6.25),

$$\text{bias}^2 = \int \left(\mathbb{E}_{\mathcal{N}_2}\left(\sum_{i=1}^{n}(x - x_i)hs'(x)\omega((x_i - x)/\sigma_1) + \frac{(x - x_i)^2}{2}hs''(x)\omega((x_i - x)/\sigma_1)\right)\right)^2 P(x)dx.$$

Let us fix a point x and use the fact that the data points on the training set are i.i.d. that follow $P(x)$, $\{x_i | i \in [1,n]\} \overset{\text{i.i.d.}}{\sim} P(x)$, to obtain:

$$\mathbb{E}_{\mathcal{N}_2}(\hat{hs}(x,\mathcal{N}_2) - hs(x)) =$$
$$\int \left(\sum_{i=1}^{n} (x-x_i) hs'(x) \omega((x_i-x)/\sigma_1) + \frac{(x-x_i)^2}{2} hs''(x) \omega((x_i-x)/\sigma_1) \right) P(x) dx$$
$$n \int \left((x-y) hs'(x) \omega((y-x)/\sigma_1) + \frac{(x-y)^2}{2} hs''(x) \omega((y-x)/\sigma_1) \right) P(y) dy.$$

By changing the variable $z = \frac{y-x}{\sigma_1}$, we have:

$$\mathbb{E}_{\mathcal{N}_2}(\hat{hs}(x,\mathcal{N}_2) - hs(x)) =$$
$$n \left(hs'(x) \int z\sigma_1 \omega(z) P(x+z\sigma_1)\sigma_1 dz + hs''(x) \int z\sigma_1 \omega(z) P(x+z\sigma_1)\sigma_1 dz \right) \quad (6.27)$$

We also consider that the probability distribution $P(x)$ is locally smooth of second order such that its Taylor approximation:

$$P(x+z\sigma_1) = P(x) + z\sigma_1 P'(x) + \frac{(z\sigma_1)^2}{2} P''(x) + o((z\sigma_1)^2),$$

makes sense. By injecting the expansion into (6.27) we obtain:

$$\mathbb{E}_{\mathcal{N}_2}(\hat{hs}(x,\mathcal{N}_2) - hs(x)) =$$
$$n\sigma_1^3\sigma_k^2 \left(hs'(x)P'(x) + 1/2.hs''(x)P(x) \right) + o((z\sigma_1)^2). \quad (6.28)$$

Therefore, we have the final expression for the bias:

$$\text{bias}^2 = (n\sigma_1^3\sigma_k^2)^2 \int \left(hs'(x)P'(x) + 1/2.hs''(x)P(x) \right)^2 P(x) dx + o((z\sigma_1)^2). \quad (6.29)$$

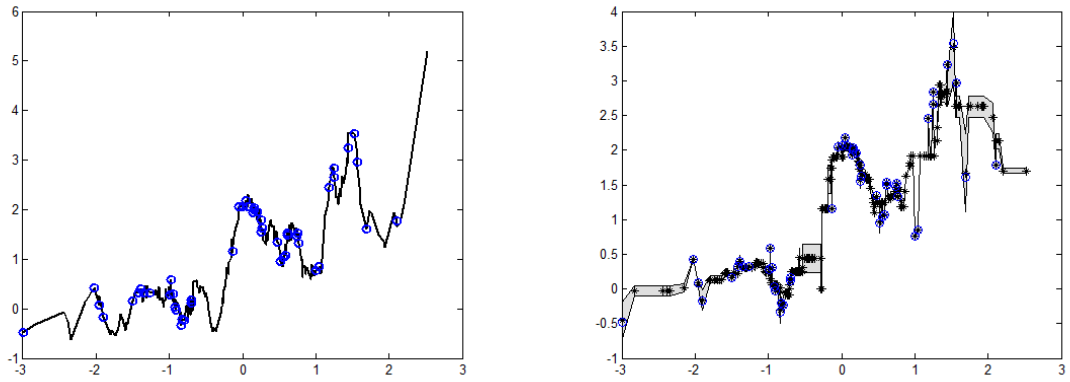
Doing the same kind of calculation for the variance, one has:

$$\begin{aligned} \text{variance} &= \int \mathbb{E}_{\mathcal{N}_2} \left[\hat{hs}(x,\mathcal{N}_2) - \mathbb{E}_{\mathcal{N}_2} \hat{hs}(x,\mathcal{N}_2) \right]^2 P(x) dx, \\ &= \int \frac{\sigma(x) R(K)}{n\sigma_1 \hat{P}(x)} P(x) dx, \end{aligned} \quad (6.30)$$

with $R(K) = \int w(u)^2 du$, $\hat{P}(x)$ is the estimate of $P(x)$ obtained thanks to the Parzen windows, $\sigma(x)$ represents the noise present on each $hs(x)$ that corrupts the data. See formula (4.11) in [138] for more details.

Concerning the smoothing parameter, we can see that when its value is small, i.e., $\sigma_1 \to 0$, so we focus on small scale details and thus we reduce the bias of estimation. In the contrary, the variance of the estimator is increased. Since by reducing σ_1, with less effect of smoothing, each of our predictions will be an average of less observations, thus they are going to vary more. In summary, this technique has a bias and we just need to find correctly appropiate σ_1 according to the compromise we want to have.

We provide in Figure 6.1 an example of Gaussian kernel regression for a 1D signal.



(a) (b)

Figure 6.1: Illustration of the signal estimation thanks to the Gaussian kernel regression approach. In (a), one can see in black the target function, and in blue the availble data or the training set, which is composed of 30 points. In (b), different training sets of 29 points have been generated based on one of 30 and we perform a regression in the space. Dark curve is the mean prediction over all the different training sets, and the grey are is the variance.

For the case of 2D images, the corresponding local neighbourhood is represented in red in Figure 6.2, thanks to it we approximate the point $hs(x)$ as a weighted mean. More precisely, we can now revisit the expression of the SEM image fusion using the AGB filter, proposed in previous chapter, from the viewpoint of kernel regression:

We can now revisit the expression of the SEM image fusion using the AGB filter, proposed in previous chapter, from the viewpoint of kernel regression:

$$\widehat{hs}(x) = \frac{1}{W_p(x)} \sum_{x_i \in E} hs(x_i)\widehat{k}(x, x_i), \tag{6.31}$$

where the AGB kernel is written as the product of two rbf kernels:

$$\widehat{k}(x, x_i) = k_{space}(x, x_i)k_{EDX+BSE}(x, x_i) \tag{6.32}$$

given by

$$k_{space}(x, x_i) = g_s(\|x_i - x\|)M(\widetilde{O}(x_i)),$$

$$k_{EDX+BSE}(x, x_i) = g_s(|\widetilde{O}(x_i) - \widetilde{O}(x)|)g_s(|r(x_i) - r(x)|),$$

with a typical normalization term:

$$W_p(x) = \sum_{x_i \in E} \widehat{k}(x, x_i), \tag{6.33}$$

M is a mask that allows us to consider just the data on the values of abundance images available at the low resolution:

$$M(\widetilde{O}(x_i)) = \begin{cases} 1, & \text{if } \widetilde{O}(x_i) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.34}$$

The function $\widetilde{O}$ was introduced in the previous chapter and represents the level of mixing of the different abundances. Finally, g_s is a Gaussian function of scale parameter s.

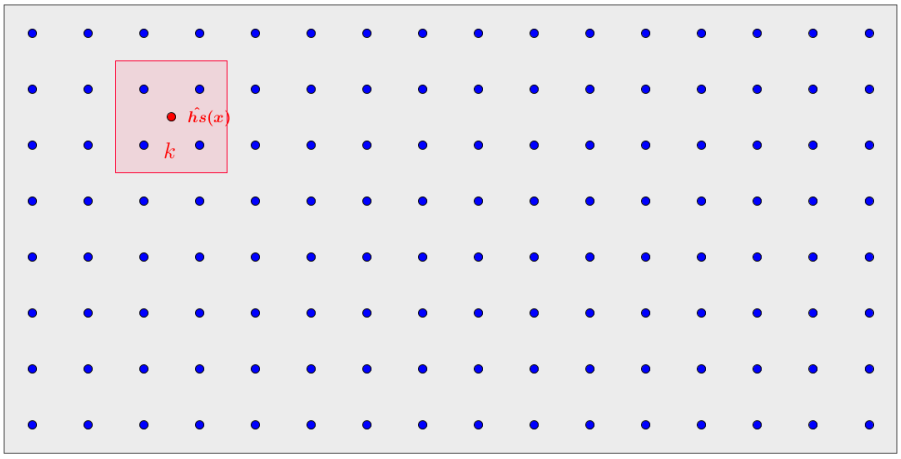


Figure 6.2: Spatial kernel regression with image information.

6.3 Gaussian kriging

Gaussian process regression [124], also called Gaussian kriging, consists in starting with a linear model regression where the value of $hs_k(x_0)$ is obtained by weighted linear combination of basis functions ϕ_i, $1 \leqslant i \leqslant N$, i.e.,:

$$\widehat{hs}(x) = \sum_{i=1}^{N} \omega_i \phi_i(x) + \epsilon, \tag{6.35}$$

where the family ϕ_i of N basis functions is used to project the data onto another feature space, and $\epsilon \sim \mathcal{N}(0, \sigma)$. By writing $\phi(x) = (\phi_1(x), \ldots, \phi_N(x))^t \in \mathbb{R}^N$ and $\omega = (\omega_1, \ldots, \omega_N)^t$, we can put the linear model in vector form as follows:

$$\hat{hs}(x) = \omega^t \phi(x) + \epsilon. \tag{6.36}$$

One can directly see the analogy with the problem of the precedent section, so we have:

$$\mathcal{P}(HS = hs(x)|x, \omega) = \mathcal{N}(\omega^t \phi(x), \sigma^2 I) \tag{6.37}$$

Let us consider a prior over the parameter ω expressing our confidence:

$$w \sim \mathcal{N}(0, S_0),$$

with $S_0 \in M_{1,N}(\mathbb{R})$. We are interested in the value of the joint distribution function of values $hs(x_1), ..., hs(x_n)$. From (6.36), we have:

$$hs(\mathcal{N}_2) \sim \mathcal{N}(0, \phi^t S_0 \phi + \sigma^2 I), \tag{6.38}$$

where $hs(\mathcal{N}_2) = (hs(x_1), \ldots, hs(x_n))^t$ and ϕ is a matrix of $M_{n,N}(\mathbb{R})$, such that $\phi(i, j) = \phi_j(x_i)$. We can then define a kernel matrix of the form:

$$K = \phi^t S_0 \phi.$$

Instead of defining basis functions, we can just work with kernels and project the data onto an infinite nonlinear feature space and then apply the representer theorem, see the theory in Chapter 1. We can see from equation (6.38):

$$cov(hs(x_i), hs(x_j)) = K(x_i, x_j) + \sigma\delta(x_i, x_j). \tag{6.39}$$

From this expression we can see that nearby data are more correlated, or in other words, the correlation between the data is governed by their distances. This lead to a concept called Gaussian process.

A Gaussian process is a statistical random function f, for which any finite subset of f taken at different spatial locations follows a multivariate Gaussian distribution and it is written as $f \sim \mathcal{GP}(0, K)$.

We consider that hs follows a Gaussian Process, which means that the covariance at two positions satisfies (6.39). Let us assume that $HS(x^*) \sim \mathcal{N}(0, K(x^*, x^*))$, which means that there is no noise on it. Moreover the joint distribution of the training outputs hs and the test outputs $hs(x^*)$ is given by

$$\begin{pmatrix} hs(\mathcal{N}_2) \\ hs(x^*) \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} K + \sigma I & K_* \\ K_* & K_{**} \end{pmatrix}\right) \tag{6.40}$$

with

$$K_* = \phi^t S_0 \phi(x^*) \text{ and } K_{**} = K(x^*, x^*).$$

Then by doing calculation, the posterior probability is obtained as:

$$\begin{aligned} &\mathcal{P}(hs(x^*)|x^*, hs((\mathcal{N}_2)), (\mathcal{N}_2)) = \\ &\mathcal{N}(K_*(K + \sigma I)^{-1} hs(x), K_{**} - K_*(K + \sigma I)^{-1} K_*^t). \end{aligned} \tag{6.41}$$

The unbiased estimator for the regression is therefore:

$$hs(x^*) = \mathbb{E}(hs|x^*, hs(\mathcal{N}_2), \mathcal{N}_2) = K_*(K + \sigma I)^{-1} hs(x). \tag{6.42}$$

We can see that the prediction is a linear combination of the observations, with a bias:

$$\text{bias} = \int \mathbb{E}_{\mathcal{N}_2}\left(\widehat{hs}(x, \mathcal{N}_2) - hs(x)\right)^2 P(x)dx. \tag{6.43}$$

We need to integrate:

$$\begin{aligned} \mathbb{E}_{\mathcal{N}_2}(\widehat{hs}(x, \mathcal{N}_2)) &= \mathbb{E}_{\mathcal{N}_2}\mathbb{E}(hs|x^*, hs(\mathcal{N}_2), \mathcal{N}_2) \\ &= \mathbb{E}(hs|X = x^*) = hs(x^*). \end{aligned}$$

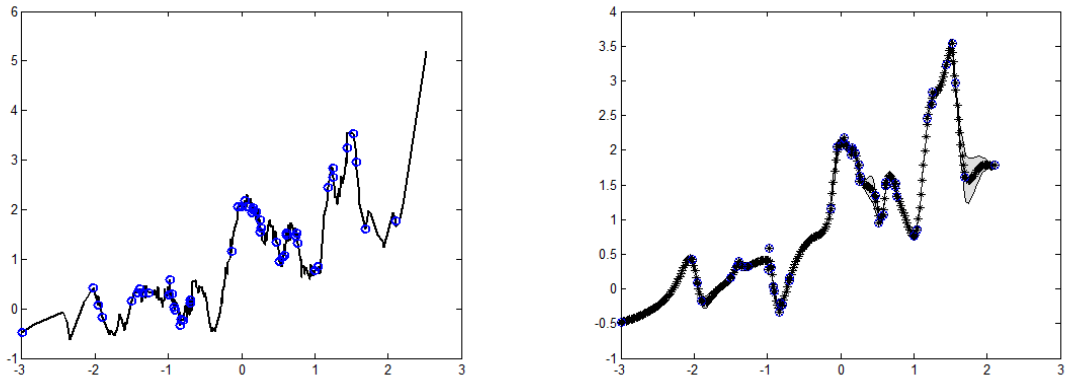
Hence, the solution has no bias. The variance is more complicated to evaluate since we do not have access to the probability of the sampling. That is why, we did not calculate it. We note that Gaussian process regression can be seen as a simple kriging applied in a Gaussian random field. This assumption is quite advantageous when the space coordinate lie on a high dimensional space since it provides an interesting way to estimate the covariance (the kernel). It is a major issue since the Gaussian process regression results depend on the choice of covariance model. In practice, instead of fixing the covariance function, it is commonly assumed to use a parametric family of functions $k(x_i, x_j, \theta) = k_\theta(x_i, x_j)$, and then try to infer θ. A

technique used in the community of machine learning to learn the hyperparameters θ are based on maximizing the log likelihood function of $p(hs(\mathcal{N}_2)|\theta)$:

$$\log p(hs(\mathcal{N}_2)|\theta, x) = -\frac{1}{2} hs(\mathcal{N}_2)^T K_\theta^{-1} hs(\mathcal{N}_2) - \frac{1}{2}\log\det(K_\theta) - \frac{n}{2}\log 2\pi$$

and maximizing this marginal likelihood towards θ provides the complete specification of the Gaussian process hs. One can briefly note that the first term corresponds to a penalty term for a model's failure to fit observed values and the second term to a penalty term that increases proportionally to a model's complexity.

Figure 6.4 provides an illustration of Gaussian process regression for 1D signal. The kernel hyperparameters were learned to optimize the log-marginal likelihood.



(a) (b)

Figure 6.3: Illustration of the signal estimation thanks to the Gaussian process regression. In (a), one can see in black the target function, and in blue the availble data or the training set, which is composed of 30 points. In (b), different training sets of 29 points have been generated based on one of 30 and we perform a regression in the space. Dark curve is the mean prediction over all the different training sets, and the grey are is the variance.

6.4 Ordinary kriging and pansharpening fusion of information

In the Gaussian process, the model is a linear combination of the observations. The ordinary kriging [101] seeks also for this kind of mathematical assumption. It is a model based on a simple linear model regression where the value of $hs(x^*)$ is calculated as a linear combination of the other known realisations:

$$\widehat{hs}(x^*) = \sum_{i=1}^{n} \omega_i(x^*) hs(x_i).$$

However, the observed data values hs are corrupted by a Gaussian noise, it is more precise to write:

$$\widehat{hs}(x^*) = \sum_{i=1}^{n} \omega_i(x^*) hs(x_i) + \epsilon, \tag{6.44}$$

where ϵ is a Gaussian noise of zero mean and standard deviation σ. It is important to notice that the noise is independent of the spatial position. Thus we can rewrite it:

$$\mathbf{P}(HS = hs(x^*)|\omega_1(x^*), \cdots, \omega_n(x^*), hs(x_1), \ldots, hs(x_n), \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(hs(x^*) - \sum_{i=1}^{n} \omega_i(x^*) hs(x_i))^2}{2\sigma^2} \right).$$

For the sake of simplicity, let us write

$$hs(\mathcal{N}_2) = (hs(x_1), \ldots, hs(x_n)).$$

Moreover, let us consider that we have different observations of the random variable HS at position x^*. Finally, let us make the assumption that the data $\{hs_1(x^*), \ldots, hs_p(x^*)\}$ are drawn i.i.d. from the previous distribution. Then we have:

$$\mathbf{P}(hs_1(x^*), \ldots, hs_p(x^*)|\omega, hs(\mathcal{N}_2), \sigma) = \frac{1}{(\sqrt{2\pi\sigma^2})^p} \exp\left(\frac{1}{2\sigma^2} \sum_{j=1}^{p} (hs_j(x^*) - \sum_{i=1}^{n} w_i(x^*) hs(x_i))^2 \right),$$

where we have considered that $\omega = (\omega_1(x^*), \ldots, \omega_n(x^*))^t$. Taking the logarithm of the likelihood function we get:

$$\mathbf{L}(hs_1(x^*), \ldots, hs_p(x^*)|\omega, hs(\mathcal{N}_2), \sigma) = -\frac{M}{2} \log(2\pi\sigma^2) - \frac{\sigma^2}{2} \sum_{j=1}^{p} \left((hs_j(x^*) - \sum_{i=1}^{n} w_i(x^*) hs(x_i)) \right)^2. \tag{6.45}$$

Therefore, maximizing the likelihood to determine w corresponds to minimizing:

$$\widehat{\mathbf{E}}_D(w) = \sum_{j=1}^{p} \left((hs_j(x^*) - \sum_{i=1}^{n} w_i(x^*) hs(x_i)) \right)^2, \tag{6.46}$$

which represents the empirical risk function, which is an empirical variance. In fact, we would like to have the true risk which would corresponds to the limit case when $p \to \infty$. We consider that a good solution is just obtained for all potential data, thus leading us to the following true risk function:

$$\begin{aligned} \mathbf{E}_{HS|X=x^*}(w) &= \mathbb{E}\left(hs(x^*) - \sum_{i=1}^{n} w_i(x^*) hs(x_i) \right)^2 \\ &= \text{var}\left(hs_j(x^*) - \sum_{i=1}^{n} w_i(x^*) hs(x_i) \right). \end{aligned} \tag{6.47}$$

However, we do not have access to different realisations of the random field and consequently, this variance. To achieve statistical inference from a single event, the theory of geostatistics has replaced the hypothesis of independent repetitions with some assumption on the random field. The first hypothesis consists in considering that some of its characteristics are identical on all the positions, like the mean for example. Another hypothesis is to consider that the expectations of some quantities are accessible by integrals over space. These assumptions are called stationarity and ergodicity. More precisely, our random field is stationary of second order if we have:

$$\mathbf{E}_{HS}(HS(x_i)) = \mathbf{E}_{HS}(HS), \tag{6.48}$$

$$\mathrm{cov}_{HS}(HS(x_i), HS(x_j)) = \mathrm{cov}_{HS}(HS(x_j + \tau), HS(x_j)) = \mathrm{C}(\tau). \tag{6.49}$$

The ergodicity is a property which reinforces the notion of stationary and provides the almost sure convergence of spatial empirical mean to the true mean when the field "goes to infinit".The random field HS is ergodic if one has:

$$\frac{1}{|E|}\int_E (HS(x)dx) \underset{E\to+\mathbb{R}^2}{\longrightarrow} \mathbf{E}_{HS}(HS) \tag{6.50}$$

where $|E|$ represents the volume of space E, i.e., the space domain of the image.

Thanks to these hypotheses we are able to estimate the covariance of the field. In order to guarantee that this estimator is unbiased, it is necessary to add the following additional condition:

$$\sum_{i=1}^{N} w_i(x^*) = 1.$$

Then, we have:

$$\mathbf{E}_{HS}\left(\sum_{i=1}^{n} w_i(x^*)HS(x_i)\right) = \sum_{i=1}^{n} w_i(x^*)\mathbf{E}_{HS}(HS(x_i)) = \mathbf{E}_{HS}(HS(x_i)). \tag{6.51}$$

This leads to a new cost function:

$$\mathcal{L}(w) = \mathrm{var}\left(hs_j(x^*) - \sum_{i=1}^{n} w_i(x^*)hs(x_i)\right) + 2\mu\left(\sum_{i=1}^{N} w_i(x^*) - 1\right). \tag{6.52}$$

By developing (6.52), we obtain:

$$\mathcal{L}(w) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_j(x^*)w_i(x^*)C(x_j, x_i)$$
$$-2\sum_{i=1}^{n} w_i(x^*)C(x^*, x_i) + C(x^*, x^*) + 2\mu\left(\sum_{i=1}^{N} w_i(x^*) - 1\right).$$

We derive it according to w_i and μ and equal to zero, and we obtain the following system:

$$\begin{cases} 2\sum_{j=1}^{n} w_j(x^*)C(x_j, x_i) - 2C(x^*, x_i) - 2\mu = 0 \quad \forall i \in [i, n] \\ \\ \sum_{j=1}^{N} w_j(x^*) = 1 \end{cases} \tag{6.53}$$

which in matricial form, and using (6.49), it leads to the following problem:

$$\begin{pmatrix} C(x_1 - x_1) & \dots & C(x_1 - x_n) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C(x_1 - x_n) & \dots & C(x_n - x_n) & 1 \\ 1 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} w_1(x^*) \\ \vdots \\ w_n(x^*) \\ -\mu \end{pmatrix} = \begin{pmatrix} C(x_1 - x^*) \\ \vdots \\ C(x_n - x^*) \\ 1 \end{pmatrix} \tag{6.54}$$

This system of equations is the same than the one of the Gaussian process except for the term μ. This last term is quite important in our case, since it guarantees that the data are on the simplex, which is the case with our abundance maps, and consequently is more physically plausible. But the fundamental difference is that in Gaussian process, the estimated parameters are those optimize the posterior of the regression of the validation set. In geostatistics, the property of stationarity of the field is the key in the optimization.

Let us define a function called the variogram:

$$\gamma(\tau) = \frac{1}{2}\text{var}\left(HS(x) - HS(x+h)\right), \tag{6.55}$$

which summarizes the variations of the variance of a spatial field according to the translations, i.e., given a distance and a direction. It satisfies the following properties

$$\begin{aligned} \gamma(\tau) &\geq 0, \\ \gamma(\tau) &= C(0) - C(\tau). \end{aligned}$$

Instead of solving the kriging with the covariance, one can therefore do it with the variogram. The advantage of working with the variogram is we do not lean on the estimation of the mean. Morerover one can define the empirical variogram by:

$$\hat{\gamma}(\tau) = \frac{1}{2\,N_\tau} \sum_{i=1}^{N_\tau} \|HS(x_i) - HS(x_i + \tau)\|_2^2, \tag{6.56}$$

where N_τ is the number of elements distant τ. However, working with the empirical variogram does not give satisfactory esults. First, because this function is not necessary negative definite, we might not have a global minimum of the kriging functional. The second point is that we want to have results with a generalization power, thus avoiding an overfitting to the available data. To overcome this issue the basic idea is to fit a model of variogram. There are different admissible parametric variograms,

- exponential:
$$\gamma(\tau) = C\left(1 - e^{-\frac{\tau}{a}}\right), \tag{6.57}$$
- Gaussian:
$$\gamma(\tau) = C\left(1 - e^{-\left(\frac{\tau}{a}\right)^2}\right), \tag{6.58}$$
- spherical:
$$\gamma(\tau) = \begin{cases} C\left(\frac{3}{2}\frac{\tau}{a} - \frac{1}{2}\left(\frac{\tau}{a}\right)^3\right) & \text{if } 0 \leqslant \tau \leqslant a \\ C & \text{if } \tau \geqslant a \end{cases}, \tag{6.59}$$

- linear:
$$\gamma(\tau) = C\frac{\tau}{a}, \tag{6.60}$$
- power:
$$\gamma(\tau) = C\left(\frac{\tau}{a}\right)^b, \text{ where } 0 < b \leq 2. \tag{6.61}$$

In our case, in order to be consistent with the rest of the chappter, we adopted the Gaussian variogram, such that its parameters to be fit are a and C. In addition, there is another parameter which is defined as

$$C_0 = \lim_{\tau \to 0} \gamma(\tau),$$

called the nugget parameter, and it represents the amount of variance not explained by the model that we have chosen. In order to estimate the parameters, that let us as write as hyperparameter θ, there are different techniques. The simple one that we consider here consists in taking the hyperparameter θ^* that minimizes the following functional:

$$\theta^* = \arg\min \sum_{k=1}^{K} (\gamma_\theta(\tau_k) - \hat{\gamma}(\tau_k))^2, \tag{6.62}$$

where the experimental variogram is evaluated at K distances τ_k. Contrary to the previous section, here we take advantage of the geometrical information, which is crucial in a low dimensional field.

Using the experimental variogram as $\gamma_{\theta^*}(\tau)$, we can now write the system to perform the kriging estimation:

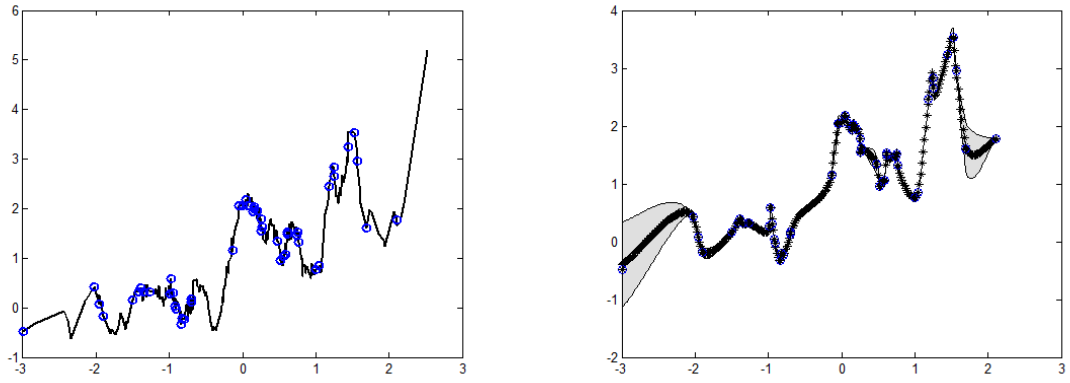
$$\begin{pmatrix} 0 & \gamma_{\theta^*}(x_1 - x_2) & \dots & \gamma_{\theta^*}(x_1 - x_n) & 1 \\ \gamma_{\theta^*}(x_1 - x_2) & 0 & \dots & \gamma_{\theta^*}(x_2 - x_n) & 1 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ \gamma_{\theta^*}(x_1 - x_n) & \dots & \gamma_{\theta^*}(x_{n-1} - x_n) & 0 & 1 \\ 1 & \dots & \dots & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} w_1(x^*) \\ w_2(x^*) \\ \vdots \\ w_n(x^*) \\ -\mu \end{pmatrix} = \begin{pmatrix} \gamma_{\theta^*}(x_1 - x^*) \\ \gamma_{\theta^*}(x_2 - x^*) \\ \vdots \\ \gamma_{\theta^*}(x_n - x^*) \\ 1 \end{pmatrix} \tag{6.63}$$

Let us come back now to our problem of multimodal pansharpening and how to inject the information of the BSE image r into the multispectral hs.

First, we need to impose the assumption that the information is locally stationary of second order. The idea consist in computing a partition of the image domain E into "homogeneous" classes, called in image processing, superpixels. In particular, we have used the SLIC algorithm for superpixels, which is accepted in the state-of-the-art as good algorithm. Then, on each of this superpixel, we perform a local kriging of the abundance maps HS. Let us write $SP(x^*)$ the super pixel containing x^*. Hence, our formulation of the local kriging is :

$$\mathcal{L}(w) = \text{var}\left(hs_j(x^*) - \sum_{i \in SP(x^*)} w_i(x^*) hs(x_i)\right) + 2\mu\left(\sum_{i=1}^{N} w_i(x^*) - 1\right). \tag{6.64}$$

This new formulation 6.64 does not change the equation system, we just focus on the data near the positions x^*, where the locality is added thanks to the backscattered image. We note other works that overcome the issue of using kriging with huge quantity of data [105, 112, 141].



(a) (b)

Figure 6.4: Illustration of the signal estimation thanks to the kriging regression. In (a), one can see in black the target function, and in blue the availble data or the training set, which is composed of 30 points. In (b), different training sets of 29 points have been generated based on one of 30 and we perform a regression in the space. Dark curve is the mean prediction over all the different training sets, and the grey are is the variance.

Second, the approach to introduce the morphological information of BSE image r into the pansharpening of hs consists in adding a regularization term that would inject this information. A regularization that is often considered on regression-like is a Bayesian regularization with a Gaussian prior which is linked to the ridge regularization. Here we use a Laplace prior on ω which is related to the Lasso regularization which depends on r. Namely, the distribution of the prior is

$$\mathcal{P}(\omega|k,\lambda) = K \prod_{j=1}^{n} \lambda C_r(x^*, x_j) \exp\left(-\lambda C_r(x^*, x_j) w_j(x^*)\right),$$

where C_r is a covariance function of the BSE image, and K a constant of normalization. Thus, we can rewrite the posterior as:

$$\begin{aligned} &\mathcal{P}(\omega|hs_1(x^*),\ldots,hs_p(x^*), C_r, \lambda, hs(\mathcal{N}_2), \sigma) = \\ &\frac{\mathcal{P}(hs_1(x^*),\ldots,hs_p(x^*)|\omega, hs(\mathcal{N}_2), \sigma)\,\mathcal{P}(\omega|C_r,\lambda)}{\mathcal{P}(hs_1(x^*),\ldots,hs_p(x^*))}. \end{aligned} \tag{6.65}$$

The logarithm of the posterior distribution is given by:

$$\begin{aligned} &\log(\mathcal{P}(w|Z_0, Z, \sigma, k, \lambda)) = \\ &\text{const} - \frac{\sigma^2}{2} \sum_{j=1}^{p} \left((hs_j(x^*) - \sum_{i=1}^{n} w_i(x^*) hs(x_i)) \right)^2 - \lambda \sum_{i=1}^{n} C_r(x^*, x_i) w_i(x^*), \end{aligned}$$

where const represents a constant that does not depend on ω.

By taking now $p \to \infty$, it leads to the following true risk function:

$$\mathbf{E}_{HS|X=x^*}(w) = \text{var}\left(hs_j(x^*) - \sum_{i=1}^{n} w_i(x^*)hs(x_i)\right) \tag{6.66}$$

$$-2\lambda \sum_{j=1}^{n} C_r(x^*, x_j)w_j(x^*), \tag{6.67}$$

We need again the convex constrain: $\sum_{j=1}^{N} w_j(x^*) = 1$. Therefore, the use of Lagrange multiplier theorem provides us:

$$\mathcal{L}(w) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_j(x^*)w_i(x^*)C(x_j, x_i) - 2\sum_{i=1}^{n} w_i(x^*)C(x^*, x_i)$$
$$+C(x^*, x^*) - 2\mu(\sum_{i=1}^{N} w_i(x^*) - 1) - 2\lambda \sum_{i=1}^{n} C_r(x^*, x_i)w_i(x^*)$$

which, deriving as previously according to w_i and μ, involves the following kriging system:

$$\begin{cases} \sum_{j=1}^{n} w_j(x^*)C(x_j, x_i) - 2C(x^*, x_i) - 2\mu - 2\lambda C_r(x^*, x_i) = 0 \quad \forall i \in [i, n] \\ \\ \sum_{j=1}^{N} w_j(x^*) = 1 \end{cases} \tag{6.68}$$

or in matrix form:

$$\begin{pmatrix} C(x_1 - x_1) & \dots & C(x_1 - x_n) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C(x_1 - x_n) & \dots & C(x_n - x_n) & 1 \\ 1 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} w_1(x^*) \\ \vdots \\ w_n(x^*) \\ -\mu \end{pmatrix} = \tag{6.69}$$

$$\begin{pmatrix} C(x_1 - x^*) - \lambda C_r(x^*, x_1) \\ \vdots \\ C(x_1 - x^*) - \lambda C_r(x^*, x_1) \\ 1 \end{pmatrix} \tag{6.70}$$

where μ is a Lagrange parameter. For the BSE image, we also use Gaussian model for the covariance:

$$C_r(h) = \exp\left(-\frac{h}{a}\right),$$

where a should be fit from the empirical covariance. The parameter λ is fundamental in this (multimodal) regularized kriging since it represents the quantity of spatial information of r to be injected into hs.

6.5 Results and discussion

6.5.1 Evaluation criteria of pansharpening algorithms

The criteria used to evaluate the quality of the merged information are those conventionally considered in the context of the pansharpening literature. We will use

exactly the same criterion that we developed in the previous chapter. More precisely, we considered the following criteria:

C1: **The spectral distortion** between the enhanced multispectral image and the real multispectral image at the nominal resolution should be as small as possible. Or, in other terms, we would like to find the same materials for each pixel as the original image at high resolution;

C2: **The spatial distortion** between the enhanced multispectral image and the real one should not be too high.

C3: **The injected information.** Since the information provided by the various modalities can be relatively different, we would like to inject the useful one to improve segmentation and characterisation of the EDX image.

Let us write $\widehat{\widehat{HS}}$ the multispectral abundance EDS image at the same resolution that R, that has been provided by the sensor at high resolution. In a way it is the ground truth such that our enhanced images $\widehat{HS}$ should be compared to $\widehat{\widehat{HS}}$. Moreover, we denote by $\widehat{\widehat{\mathcal{M}}} \in M_{n,D}(\mathbb{R})$ and $\widehat{\mathcal{M}} \in M_{n,D}(\mathbb{R})$ the two matrices representing respectively $\widehat{\widehat{HS}}$ and $\widehat{HS}$, where n is the total number of pixels (i.e., $n = N_1 \times N_2$), and D the number of abundance maps. We write by $\widehat{\mathcal{M}}_{i,:}$, $\forall i \in [1, n]$, a spectra of $\widehat{HS}$, and $\widehat{\mathcal{M}}_{:,k} \forall k \in [1, D]$ a map of $\widehat{HS}$. We have now all the notations to introduce the five parameters.

Cross correlation (CC) is a measure that evaluates the spatial distortion defined as:

$$CC(\widehat{HS}, \widehat{\widehat{HS}}) = \frac{1}{D} \sum_{k=1}^{D} CCS(\widehat{\mathcal{M}}_{:,k}, \widehat{\widehat{\mathcal{M}}}_{:,k}), \tag{6.71}$$

where:

$$CCS(\widehat{\mathcal{M}}_{:,k}, \widehat{\widehat{\mathcal{M}}}_{:,k}) = \frac{(\sum_{i=1}^{n} \widehat{\mathcal{M}}_{i,k} - \mu_{\widehat{\mathcal{M}}_{:,k}})(\sum_{i=1}^{n} \widehat{\widehat{\mathcal{M}}}_{i,k} - \mu_{\widehat{\widehat{\mathcal{M}}}_{:,k}})}{\sqrt{\sum_{i=1}^{n} (\widehat{\mathcal{M}}_{i,k} - \mu_{\widehat{\mathcal{M}}_{:,k}})^2 \sum_{i=1}^{n} (\widehat{\widehat{\mathcal{M}}}_{i,k} - \mu_{\widehat{\widehat{\mathcal{M}}}_{:,k}})^2}},$$

with $\mu_{\widehat{\mathcal{M}}_{:,k}} = n^{-1} \sum_{i=1}^{n} \widehat{\mathcal{M}}_{i,k}$ being the empirical mean. The CC is optimal when it is close to 1.

Spectral Angle Mapper (SAM) is a measure that assesses the spectral distortion by computing

$$SAM(\widehat{HS}, \widehat{\widehat{HS}}) = \frac{1}{n} \sum_{k=1}^{n} \widetilde{SAM}(\widehat{\mathcal{M}}_{i,:}, \widehat{\widehat{\mathcal{M}}}_{i,:}), \tag{6.72}$$

with

$$\widetilde{SAM}(\widehat{\mathcal{M}}_{i,:}, \widehat{\widehat{\mathcal{M}}}_{i,:}) = \arccos\left(\frac{< \widehat{\mathcal{M}}_{i,:}, \widehat{\widehat{\mathcal{M}}}_{i,:} >}{\|\widehat{\mathcal{M}}_{i,:}\| \|\widehat{\widehat{\mathcal{M}}}_{i,:}\|} \right),$$

where $< \cdot, \cdot >$ is the Euclidean inner product of vectors associated to the norm L^2, and where $\| \cdot \|$ is the L^2 norm of vectors. The SAM is optimal when it is near to 0.

Root mean squared error (RMSE) measures the mean residual error of fusion and is defined as:

$$RMSE(\widehat{HS}, \widehat{\widehat{HS}}) = \frac{\|\widehat{\mathcal{M}} - \widehat{\widehat{\mathcal{M}}}\|_F}{n.D}, \tag{6.73}$$

where $\| \cdot \|_F$ is the Frobenius norm of a matrix A, i.e., $\|A\|_F = \sqrt{\text{trace}(AA^t)}$. The RMSE is optimal when it is near to zero.

Synthetic adimensional global error (ERGAS) offers a global measure of the quality of an enhanced image. It is given by the expression:

$$ERGAS(\widehat{HS}, \widehat{\widehat{HS}}) = \sqrt[100d]{\sum_{k=1}^{D} \left(\frac{RMSE(\widehat{\mathcal{M}}_{:,k}, \widehat{\widehat{\mathcal{M}}}_{:,k})}{\mu_{\widehat{\mathcal{M}}_{:,k}}} \right)^2},$$

where d is the ratio between the linear resolution of the BSE image R and the abundances EDS image HS, i.e.,

$$d = \frac{R\text{-linear spatial resolution}}{HS\text{-linear spatial resolution}}.$$

The ERGAS is optimal when close to 0.

Cross correlation gradient (CC_g) is a measure that evaluates the spatial distortion defined as

$$\text{CC}_\text{g}(R, \widehat{\widehat{HS}}) = \frac{1}{D} \sum_{k=1}^{D} CCS(\mathcal{R}g, \widehat{\widehat{\mathcal{M}}}g_{:,k}),$$

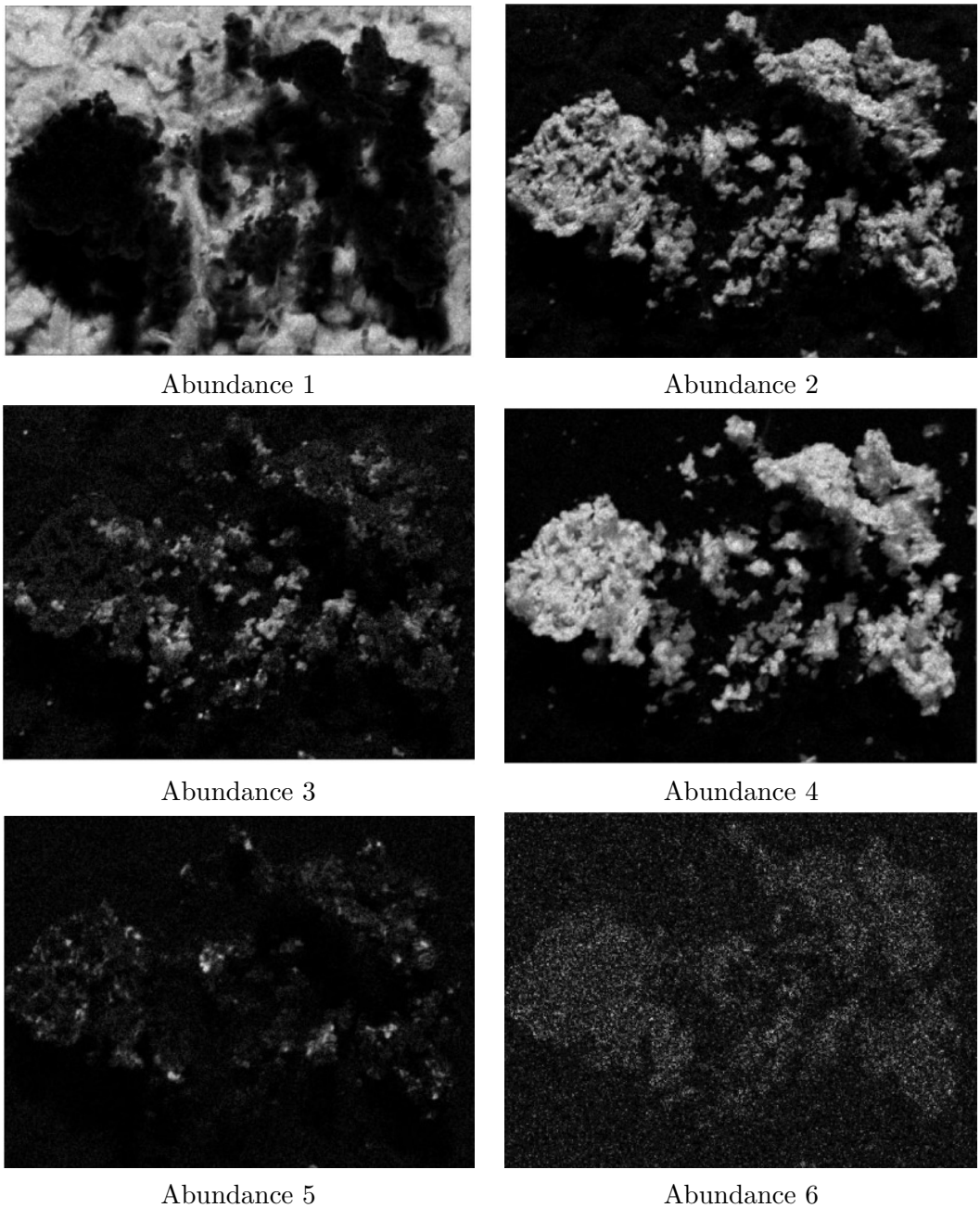
where $\mathcal{R}g$ represents the image morphological gradient [142] of the BSE image converted into a vector and $\widehat{\widehat{\mathcal{M}}}g_{:,k}$ is the image morphological gradient of the enhanced abundance map at the nominal scale converted also into a vector.

An optimal enhancement method would be a compromise between the criterion C3 and the C1 and C2.

6.5.2 Evaluation on dataset 1

The EDS abundance maps of dataset 1 are provided in Figure 6.5. The results of the enhanced abundances $\widehat{HS}$ obtained by the different techniques provided in Figures **??**, 6.7 6.13 and 6.9, where the abundances are visualized as RGB color images. Quantitative results for this case according to the five measures are presented in Table 6.1.

From these results, we note that AGB presents good performance but it smoothes a lots the image, while kriging techniques inject more gradually geometric/textural information from the BSE image. Then, even if based on these results it is difficult to definitely compare these techniques, we might see that the advantage of the kriging technique is it flexibility since the operator just need to select the quantity of parsimony based on the backscattered he wants to apply.



Abundance 1 Abundance 2

Abundance 3 Abundance 4

Abundance 5 Abundance 6

Figure 6.5: The six EDS abundance maps of SEM dataset 1 at the nominal resolution.

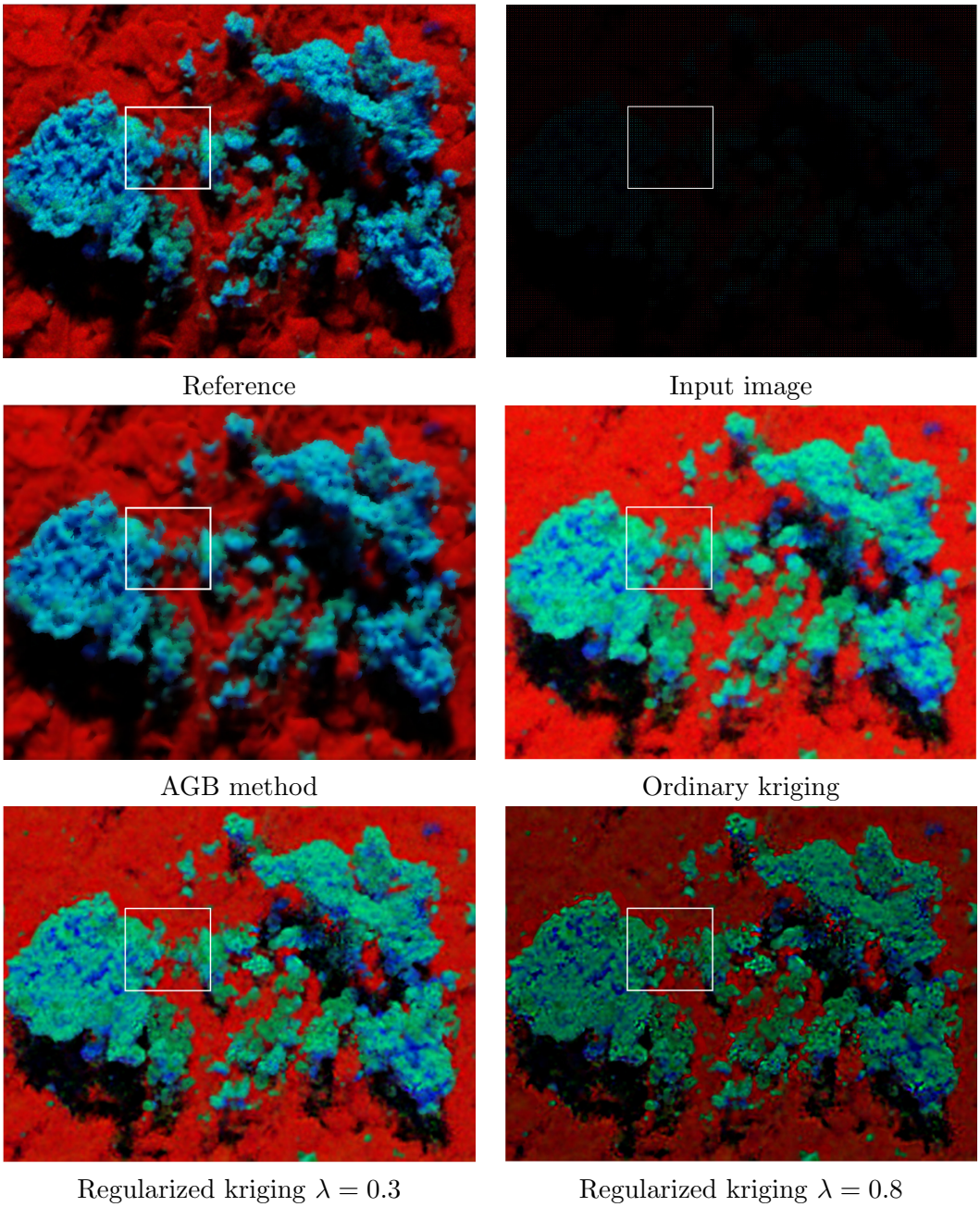


Figure 6.6: RGB color image from abundances 1,2,4 of the dataset 1 for the different spatial interpolation techniques.

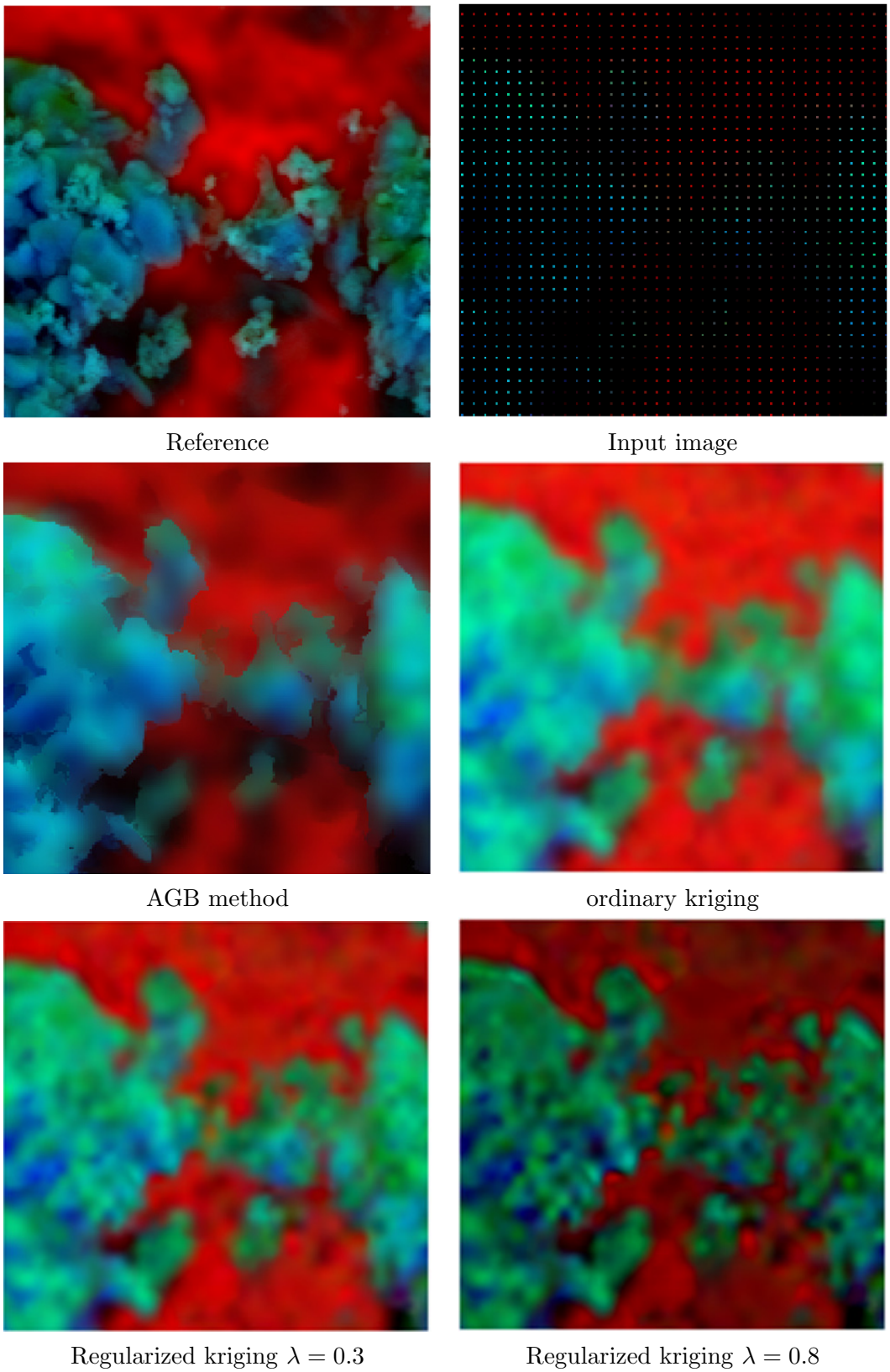


Figure 6.7: Zoom of the RGB color image from abundances 1,2,4 of the dataset 1 for the different spatial interpolation techniques.

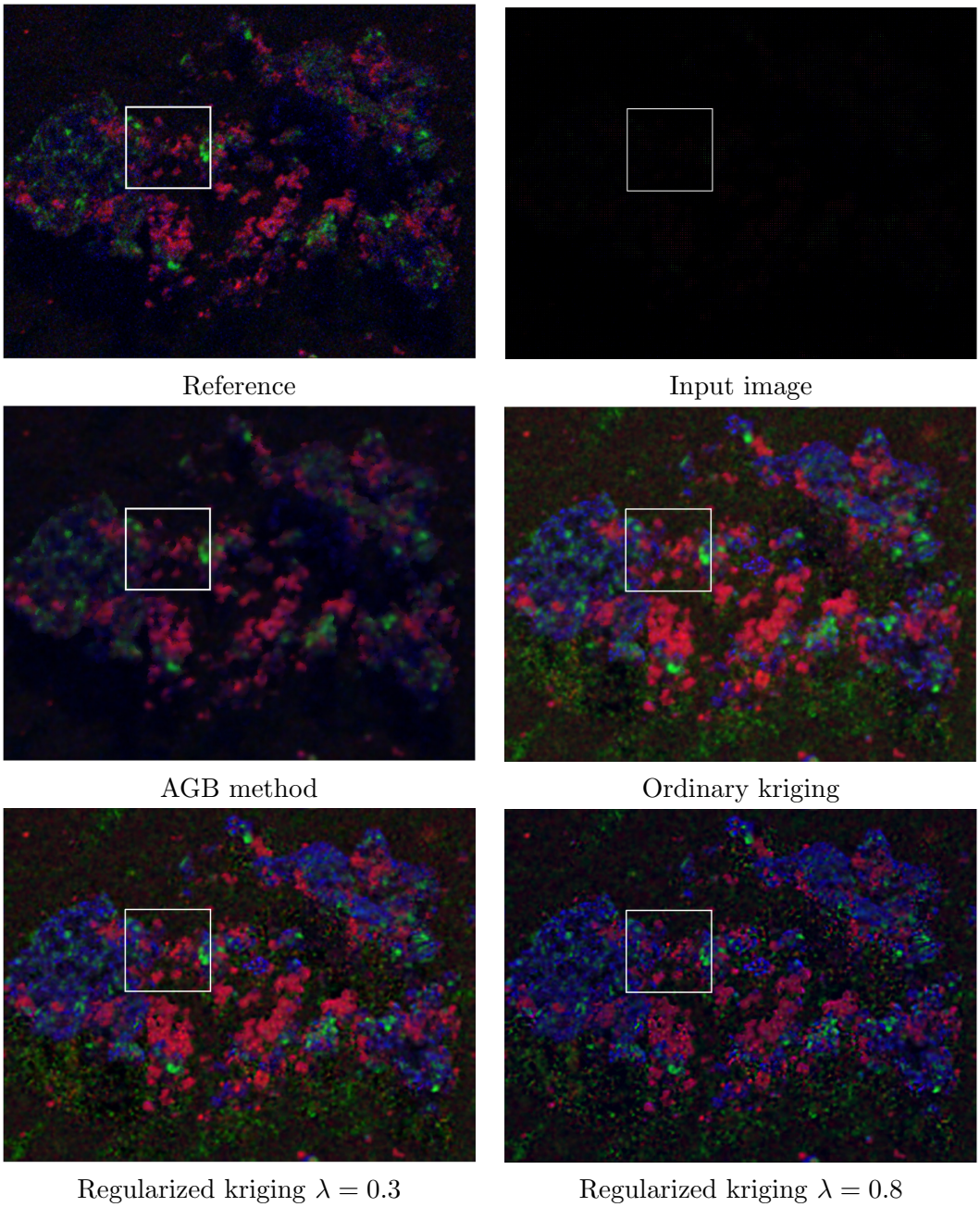


Figure 6.8: RGB color image from abundances 3,5,6 of the dataset 1 for the different spatial interpolation techniques.

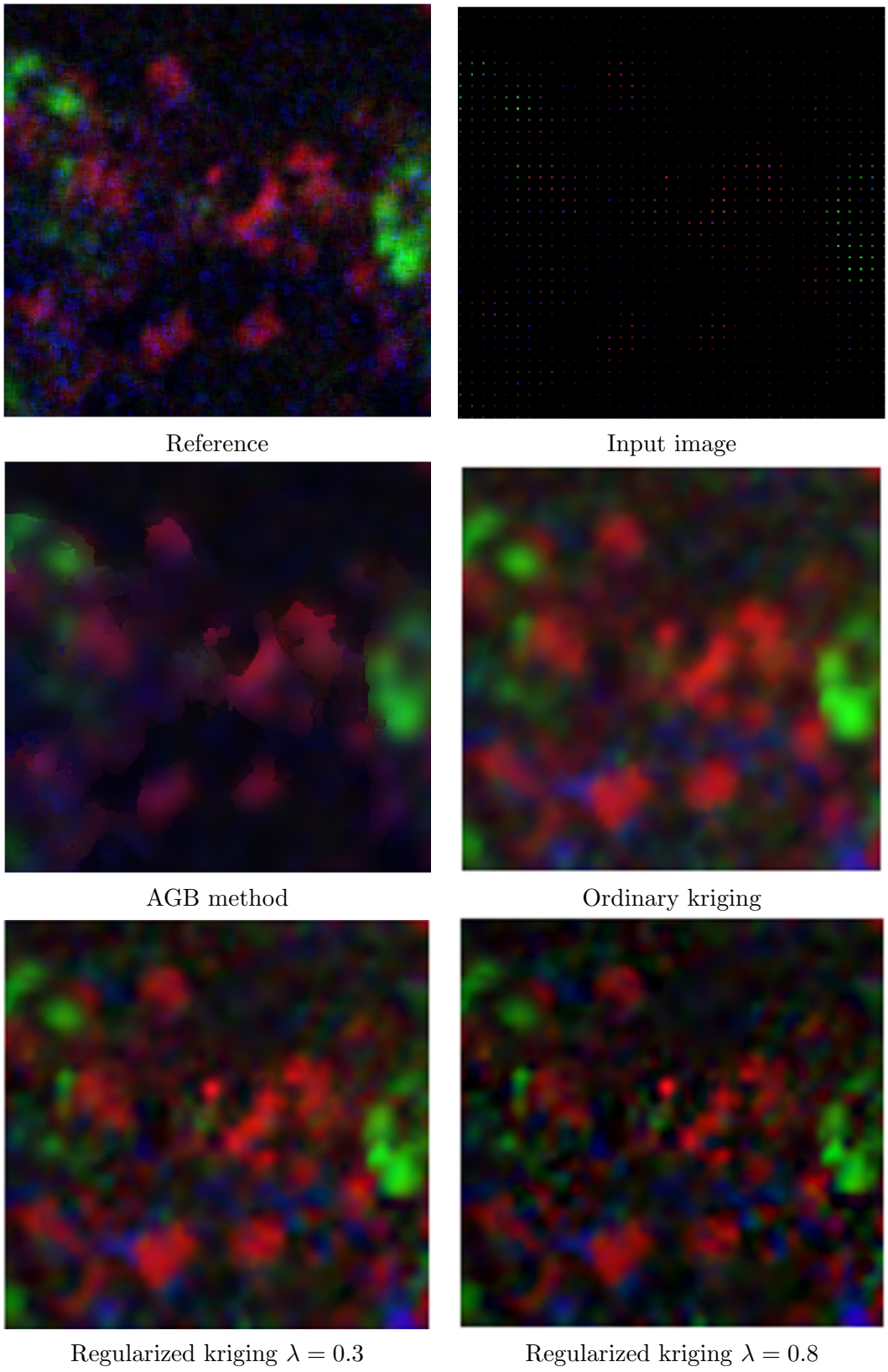


Figure 6.9: Zoom of the RGB color image from abundances 3,5,6 of the dataset 1 for the different spatial interpolation techniques.

Techniques	**CC**	**SAM**	**RMSE**	**ERGAS**	$\mathbf{CC_g}$
AGB	**0.90**	**9.5**	**11.2**	9.35	0.59
regularized kriging $\lambda = 0.3$	0.79	**8.8**	50.2	74.5	0.33
regularized kriging $\lambda = 0.6$	0.77	10.6	52.1	79.1	0.33

Table 6.1: Comparison of spatial interpolation algorithms for enhanced EDS abundance images of dataset 1.

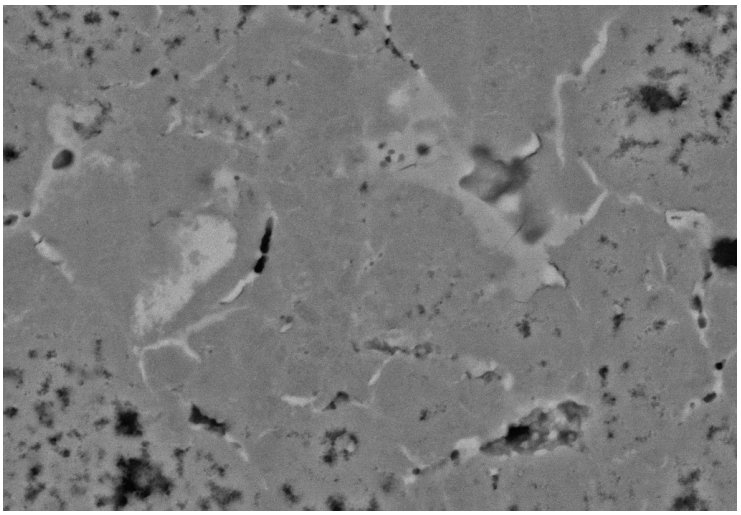


Figure 6.10: BSE image of SEM dataset 2.

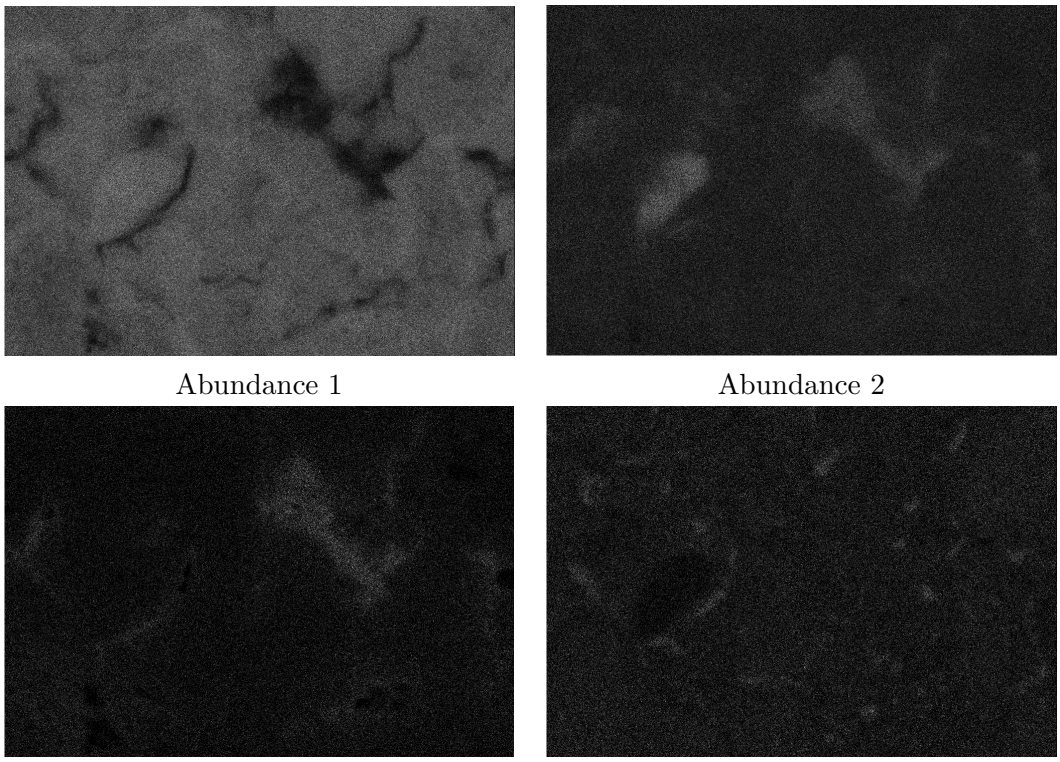
6.5.3 Evaluation on dataset 2

In the case of this multimodal SEM dataset, which has a higher level of noise, we try to inject the BSE image R, depicted in Figure 6.10, to increase the resolution of the EDS abundance maps provided in Figure 6.11, and also to denoise the images. The results for our ABG method are given in Figure 5.14.

For comparison, Figures 6.12 and 6.13 provide a visualization of the results obtained using the other methods. Quantitative results are given in Table 6.2, which lead to similar conclusions as for dataset 1.

Techniques	**CC**	**SAM**	**RMSE**	**ERGAS**	$\mathbf{CC_g}$
AGB	**0.30**	**15.7**	**1.4**	**20.1**	0.45
regularized kriging $\lambda = 0.3$	0.28	**15.7**	**1.31**	49.2	0.23
regularized kriging $\lambda = 0.6$	0.21	60.6	39.8	73.1	0.13

Table 6.2: Comparison of spatial interpolation algorithms for enhanced EDS abundance images of dataset 2.



Abundance 1 Abundance 2

Abundance 3 Abundance 4

Figure 6.11: Four EDS abundance maps of SEM dataset 2 at the nominal resolution.

6.6 Conclusion

In this chapter we reformulate the pansharpening problem in the form of a spatial regression problem. We have proposed several solutions to deal with such regression. We compared and evaluated these different solutions for the specific problem of enhanced EDS in multimodal SEM imaging. An interesting point of these two techniques is that they work on the full vector space, such that the full spectra of the multispectral image are used. The kernel regression technique preserves the positivity of the data too. The latter is physically important since the data we are dealing with represents quantities of materials. The kriging technique provides results that are visually interesting.

It appears that the technique based on regularized kriging seems to be easier to use since there is just a single parameter to tune that corresponds to the quantity of the high spatial resolution information to be injected. The technique based on kernel regression seems to have better results on the test datasets, with the criteria that we used. However this technique have four parameters, which are on the one hand, the size of the kernels, and on the other hand the three scaling parameters σ of each rbf kernel. In order to improve this technique, it might be interesting to try to learn some of these parameters automatically.

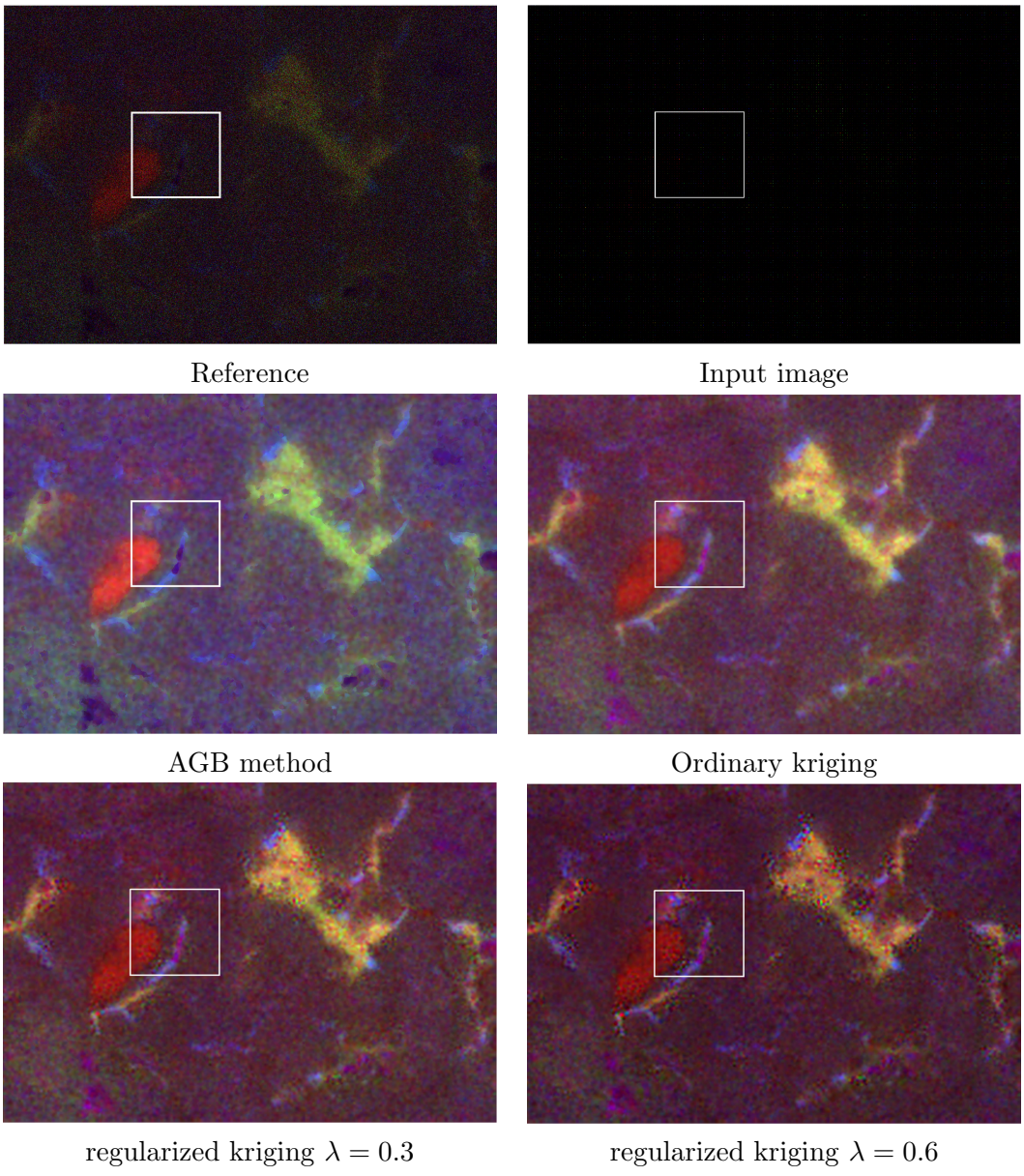


Figure 6.12: RGB color image from abundances 2,3,4 of the dataset 2 for the different spatial interpolation techniques.

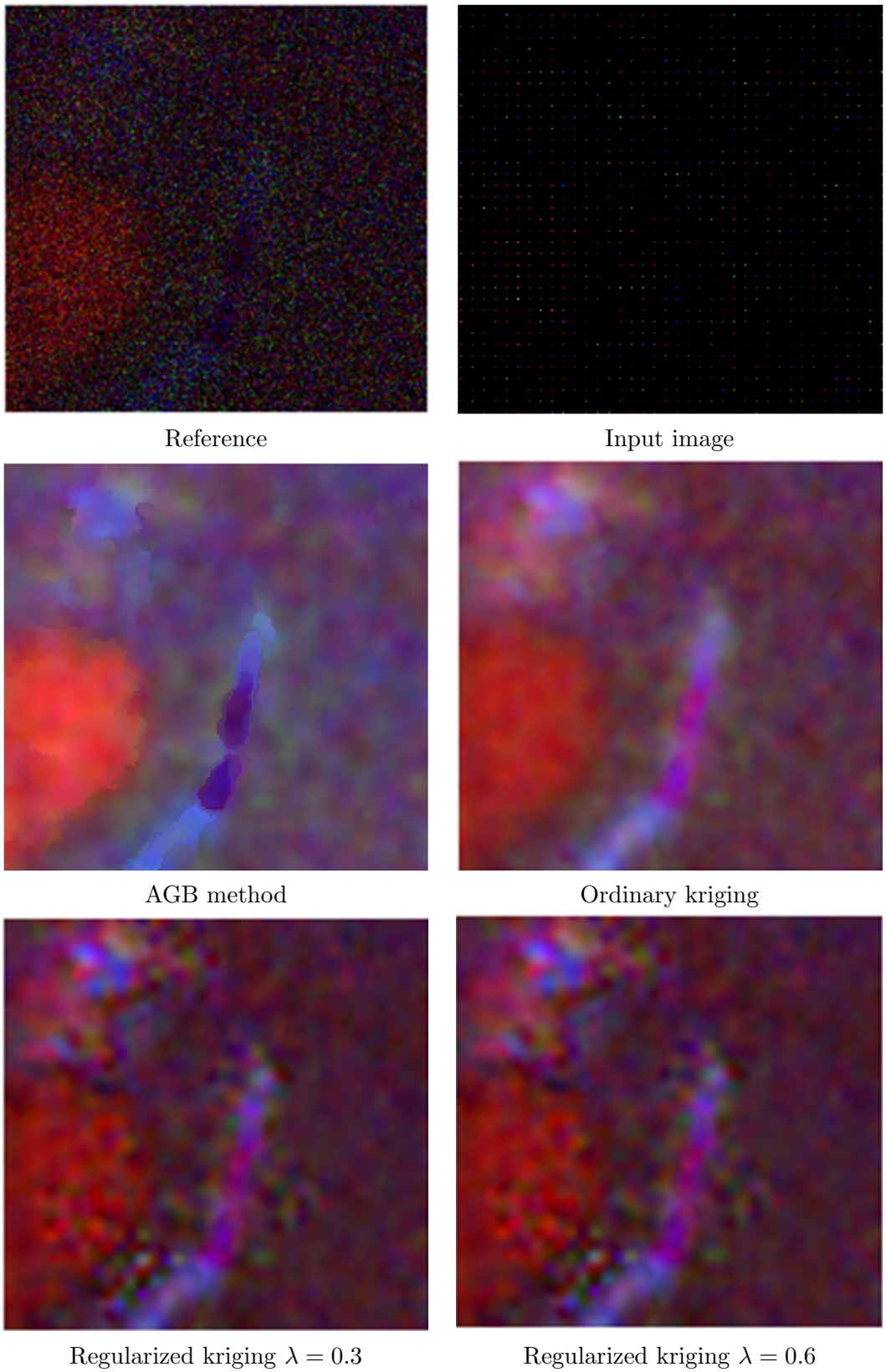


Figure 6.13: Zoom of the RGB color image from abundances 2,3,4 of the dataset 2 for the different spatial interpolation techniques.

Part IV

Conclusion

Chapter 7

Conclusions and Perspectives

7.1 Summary of main contributions

In this thesis we addressed the problem of finding relevant representations for multivariate and multimodal images.

The idea that we developed for this kind of representations is to include spatial information in the statistical learning process. We have in particular worked on multivariate spatial dimensionality reduction, spatial classification and multimodal spatial regression.

Hence, the main contributions of this thesis are just based on an appropiate use of spatial information in machine learning techniques in order to fight the curse of dimensionality on the manifold of multivariate image pixels.

We have use methods and algorithms from mathematical morphology, wavelet and scattering transform, geostatistics, regression and kernel-based machine learning. We compared the proposed techniques theoretically and also in terms of experimental results with state-of-the-art methods.

To summarize, the contributions of our work include the following points.

Chapter 3. We have shown how to deal with spatial information in the process of multi/hyper-spectral image dimensionality reduction thanks to mathematical morphology operators. The representation techniques that we introduced in this chapter are based on the notion of the MorphPCA. Dimensionality reduction is done on the space of the data without any approximation regarding the notion of morphological covariance V_{Morpho}, where V_{Morpho} is a covariance handling the morphological/spatial relationships between the image bands of the multi/hyper-spectral image. We proposed different alternatives to compute these covariance matrices. Globaly, the technique is simple in terms of computation and memory storage. Moreover, we also proposed some criteria to assess the quality of the image representation after the dimensionality reduction. Some of them are based on mathematical morphology, namely the 3D pattern spectrum and the $\alpha-$flat zone partition, and are used to evaluate if the reconstructed image preserves globally and locally the similarity to the original hyperspectral image. Finally, we also consider the interest of MorphPCA for supervised classification on the reduced data. According to the entire set of criteria, adding spatial information improves the dimensionality reduction. However, as we have shown, a good dimensionality reduction is

obtained when combining spatial and spectral feature spaces.

Chapter 4. We have addressed the interest of spatial information in the process of multi/hyper-spectral image classification. To perform this classification of pixels, we implemented a deep convolutional descriptor. This descriptor is based on two innovative techniques called scattering transform and kernel mean map. Thanks to it, a translation-invariant representation of the texture is obtained, which is also Lipschitz stable to deformations. To evaluate this descriptor, we just calculated the mean of the scattering coefficients provided by each layer embedded on the Hilbert space of a rbf kernel. This classifier simplifies convolutional neural network since fewer parameters are needed. This descriptor constitutes an interesting way to perform deep learning when the training data set is limited. We studied the statistical and pattern recognition properties of this descriptor, and prove also empirically that it presents a good classification performance.

Chapters 6 and 7. We have studied how spatial information is naturally used on a process of fusion of information called pansharpening. To perform this fusion of information of multimodal SEM images, different techniques were deeply considered. One of the major difficulties of this part was the lack of literature on this kind of techniques applied to multimodal SEM. Thus, our starting point was the pansharpening state-of-the-art solutions proposed for multimodal remote sensing images. We have developed three techniques. The first one consists in using the nonparametric kernel regression, where the kernel handles spatial and spectral information. One of these well known kernels widely used in image processing is the bilateral filter. This approach presents a good compromise according to the criteria of evaluation that we used. We proposed also to use a technique based on morphological wavelets for upsampling. Finally, the third approach is based on ordinary kriging. From our viewpoint, the last is quite interesting since we have a way to measure how much of the second modality we want to inject into the first one. However, the inconvenience with such a technique is that we lose some physical properties of the final results, even if visually the results seem better.

7.2 Suggestions for future work

We finally discuss some open questions and suggestions based on the works we proposed.

Chapter 3: The following points deserve some developments,

- The MorphPCA dimensionality reduction techniques proposed were used on hyperpectral images. It might be interesting to consider their interest for other kind of multivariate images, including also multimodal ones.
- Dimensionality reduction is used to denoise images sequences, and thus investigating the interest of MorphPCA approaches might be interesting in time series.
- Some fusion of information techniques are based on the PCA, MorphPCA could replace the PCA on these techniques.

- It might be interesting to theoretically link MorphPCA with kriging, in order to have a geostatistical theory of spatial dimensionality reduction.

Chapter 4: The potential of this part of the thesis is important and different aspects should be investigated in ongoing work.

- To use other classification techniques than the SVM. We would suggest to apply Gaussian process, since it is linked with kriging and with the scattering.
- To focus more on learning the perfect Hilbert space where the scattering transform is embedded.
- To study the interest of this descriptor for the simulation of hyper/multi–spectral textures.
- To use this descriptor to solve problems other than classification.
- In the process of estimating the rbf feature space, we need to project the data onto a random space. Estimating this space in a non-random way might improve the results. By doing that, we might not emmbed the data on the rbf feature space and we can expect an improvement of the precision.
- Another interesting solution might be to use these descriptors as local visual descriptors (replacing SIFT for example), and use them in typical computer vision tasks.
- Instead of using a Gaussian average, one could use a perceptron algorithm, and learn the coefficient of the weights, closer to the convolutional neural network paradigm.

Chapters 6 and 7. Related to the problem of image fusion and pansharpening, we suggest for future investigations to work on:

- To assess the interest of using other SEM modalities, e.g., the secondary electron image, in the process of image fusion;
- To study how to improve the selection of the EDX pixels images carefully, so that when we want to increase the dimension, the most important information is used;
- To study if there exists a link between spatial regression and with compressive sensing theory;
- To develop a kriging model with an additional constraint of nonnegative coefficients (nonnegative kriging).

Bibliography

[1] http://cimewww.epfl.ch/people/buffat/Docu_ME_02_03/030116_pdf/SEM_bloc_part2.pdf. 105

[2] http://deeplearning.net/tutorial/lenet.html. 40

[3] http://www4.nau.edu/microanalysis/microprobe-sem/instrumentation.html. 103

[4] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006. 30

[5] J. Andén and S. Mallat. Multiscale scattering for audio classification. In *ISMIR*, pages 657–662, 2011. 74

[6] J. Angulo. Morphological bilateral filtering. volume 6, pages 1790–1822. Society for Industrial and Applied Mathematics, 2013. 123

[7] J. Angulo. (max, min)-convolution and mathematical morphology. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 485–496. Springer, 2015. 79

[8] F. J. Anscombe. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948. 107

[9] C. Bachmann, T. Ainsworth, and R. Fusina. Exploiting manifold geometry in hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(3):441–454, March 2005. 44

[10] A. Baddeley. Errors in binary images and an lp version of the hausdorff metric. *Nieuw Archief voor Wiskunde*, 10(4):157–183, 1992. 55

[11] R. Bauer and R. Rick. Computer analysis of x-ray spectra (eds) from thin biological specimens. *X-Ray Spectrometry*, 7(2):63–69, 1978. 111

[12] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001. 6

[13] R. E. Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015. 3, 73

[14] A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004. 83

[15] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995. 38

[16] C. M. Bishop. Pattern recognition. *Machine Learning*, 128, 2006. 30, 32, 36, 38, 40, 77, 142

[17] L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998. 38

[18] J. Bruna and S. Mallat. Invariant scattering convolution networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1872–1886, 2013. 7, 74, 76, 81, 82

[19] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983. 117

[20] G. Camps-Valls and L. Bruzzone. Kernel-based methods for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(6):1351–1362, 2005. 62

[21] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla. Composite kernels for hyperspectral image classification. *Geoscience and Remote Sensing Letters, IEEE*, 3(1):93–97, 2006. 56, 74, 94

[22] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. Atli Benediktsson. Advances in hyperspectral image classification: Earth monitoring with statistical learning methods. *Signal Processing Magazine, IEEE*, 31(1):45–54, 2014. 73

[23] G. Cavallaro, N. Falco, M. Dalla Mura, L. Bruzzone, and J. A. Benediktsson. Automatic threshold selection for profiles of attribute filters based on granulometric characteristic functions. In *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 169–181. Springer, 2015. 46

[24] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm. 94

[25] P. Chavez, S. C. Sides, and J. A. Anderson. Comparison of three different methods to merge multiresolution and multispectral data- landsat tm and spot panchromatic. *Photogrammetric Engineering and remote sensing*, 57(3):295–303, 1991. 115

[26] Chein-I-Chang. *Hyperspectral Imaging techniques for Spectral Detection and Classification*. Kluwer Academic / Plenum Publisher, 2003. 6

[27] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 7(6):2094–2107, 2014. 74

[28] J.-P. Chiles and P. Delfiner. *Geostatistics: modeling spatial uncertainty*, volume 497. John Wiley & Sons, 2009. 140

[29] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 6

[30] C. Cortes, M. Kloft, and M. Mohri. Learning kernels using local rademacher complexity. In *Advances in Neural Information Processing Systems*, pages 2760–2768, 2013. 91

[31] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001. 36, 98

[32] N. Cressie. The origins of kriging. *Mathematical geology*, 22(3):239–252, 1990. 140

[33] N. Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015. 140

[34] S. DABO-NIANG and J. Zoueu. Combining kriging, multispectral and multimodal microscopy to resolve malaria-infected erythrocyte contents. *Journal of microscopy*, 247(3):240–251, 2012. 140

[35] M. Dalla Mura, J. Benediktsson, B. Waske, and L. Bruzzone. Morphological attribute profiles for the analysis of very high resolution images. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(10):3747–3762, Oct 2010. 7, 74, 94

[36] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. van Kasteren, W. Liao, R. Bellens, A. Pižurica, S. Gautama, et al. Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2405–2418, 2014. 44, 63

[37] D. Decoste and B. Schölkopf. Training invariant support vector machines. *Machine learning*, 46(1-3):161–190, 2002. 79

[38] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in neural information processing systems*, page None, 2003. 77

[39] D. L. Donoho and C. Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003. 6, 24

[40] O. Eches. *Méthodes Bayésiennes pour le démélange d'images hyperspectrales*. PhD thesis, UNIVERSITÉ DE TOULOUSE, 2010. 14

[41] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. 22

[42] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM transactions on graphics (TOG)*, 23(3):673–678, 2004. 118

[43] Exelis. *Analyse spectrale avec ENVI*, 2012. formation : Ecole d'été SFTH. 14

[44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008. 98

[45] M. Fauvel, J. Benediktsson, J. Chanussot, and J. Sveinsson. Spectral and spatial classification of hyperspectral data using svms and morphological profiles. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(11):3804–3814, Nov 2008. 7, 51, 73, 74

[46] 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, 2013. 73

[47] S. R. Flaxman, Y.-X. Wang, and A. J. Smola. Who supported obama in 2012?: Ecological inference through distribution regression. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–298. ACM, 2015. 80, 84

[48] E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768–769, 1965. 3

[49] G. Franchi and J. Angulo. Comparative study on morphological principal component analysis of hyperspectral images. *In Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS),6th Workshop on,*, 12:1–4, 2014. 44

[50] G. Franchi and J. Angulo. Ordering on the probability simplex of endmembers for hyperspectral morphological image processing. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 410–421. Springer, 2015. 119

[51] G. Franchi and J. Angulo. A deep spatial/spectral descriptor of hyperspectral texture using scattering transform. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 3568–3572. IEEE, 2016. 74

[52] G. Franchi and J. Angulo. Morphological principal component analysis for hyperspectral image analysis. *ISPRS International Journal of Geo-Information*, 5(6):83, 2016. 44

[53] G. Franchi and J. Angulo. Morphological principal component analysis for hyperspectral image analysis. 2016. 60

[54] G. Franchi, J. Angulo, and D. Sejdinović. Hyperspectral image classification with support vector machines on kernel distribution embeddings. pages 1898–1902, Sept 2016. 74, 89

[55] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001. 5, 32, 79

[56] A. A. Goshtasby. Similarity and dissimilarity measures. In *Image Registration*, pages 7–66. Springer, 2012. 50

[57] J. Goutsias and H. J. Heijmans. Nonlinear multiresolution signal decomposition schemes. i. morphological pyramids. *IEEE Transactions on image processing*, 9(11):1862–1876, 2000. 121

[58] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B. K. Sriperumbudur. Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems*, pages 1205–1213, 2012. 93

[59] J. A. Gualtieri and R. F. Cromp. Support vector machines for hyperspectral remote sensing classification. In *The 27th AIPR Workshop: Advances in Computer-Assisted Recognition*, pages 221–232. International Society for Optics and Photonics, 1999. 73

[60] K. Guilfoyle, M. L. Althouse, and C.-I. Chang. Further investigations into the use of linear and nonlinear mixing models for hyperspectral image analysis. In *AeroSense 2002*, pages 157–167. International Society for Optics and Photonics, 2002. 44

[61] R. M. Haralick, K. Shanmugam, and I. H. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973. 75

[62] R. M. Haralick, X. Zhuang, C. Lin, and J. S. Lee. The digital morphological sampling theorem. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(12):2067–2090, 1989. 121

[63] J. He, L. Zhang, Q. Wang, and Z. Li. Using diffusion geometric coordinates for hyperspectral imagery representation. *Geoscience and Remote Sensing Letters, IEEE*, 6(4):767–771, 2009. 62

[64] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2013. 117

[65] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 40

[66] H. J. Heijmans and J. Goutsias. Nonlinear multiresolution signal decomposition schemes. ii. morphological wavelets. *IEEE Transactions on Image Processing*, 9(11):1897–1913, 2000. 121

[67] H. J. Heijmans and J. Roerdink. *Mathematical morphology and its applications to image and signal processing*, volume 12. Springer Science & Business Media, 1998. 122

[68] H. J. Heijmans and A. Toet. Morphological sampling. *CVGIP: Image understanding*, 54(3):384–400, 1991. 121

[69] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933. 115

[70] E. V. Huntington. Mathematics and statistics, with an elementary account of the correlation coefficient and the correlation ratio. *The American Mathematical Monthly*, 26(10):421–435, 1919. 50

[71] E. Izquierdo-Verdiguier, V. Laparra, L. Gómez-Chova, and G. Camps-Valls. Encoding invariances in remote sensing image classification with svm. *IEEE Geoscience and Remote Sensing Letters*, 10(5):981–985, 2013. 79

[72] A. K. J. Johnson. The stanford cs class cs231n: Convolutional neural networks for visual recognition, `http://cs231n.github.io/neural-networks-1/`. 39

[73] X. Kang, S. Li, and J. A. Benediktsson. Spectral–spatial hyperspectral image classification with edge-preserving filtering. *IEEE transactions on geoscience and remote sensing*, 52(5):2666–2677, 2014. 117

[74] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 39

[75] C. A. Laben and B. V. Brower. Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening, Jan. 4 2000. US Patent 6,011,875. 110, 116, 139

[76] A. Landström. An approach to adaptive quadratic structuring functions based on the local structure tensor. In *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 729–740. Springer, 2015. 123

[77] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 38

[78] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998. 75

[79] J. A. Lee and M. Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing*, 72(7):1431–1443, 2009. 61

[80] S. Lefevre, L. Chapel, and F. Merciol. Hyperspectral image classification from multiscale description with constrained connectivity and metric learning. pages 1–4, June 2014. 74

[81] S. Lesage, R. Gribonval, F. Bimbot, and L. Benaroya. Learning unions of orthonormal bases with thresholded singular value decomposition. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–293. IEEE, 2005. 30

[82] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000. 30

[83] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. Atli Benediktsson, and A. Plaza. Multiple feature learning for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 53(3):1592–1606, 2015. 74, 94

[84] J. Li, Y. Qu, C. Li, Y. Xie, Y. Wu, and J. Fan. Learning local gaussian process regression for image super-resolution. *Neurocomputing*, 154:284–295, 2015. 140

[85] J. Li, P. Reddy Marpu, A. Plaza, J. M. Bioucas-Dias, and J. Atli Benediktsson. Generalized composite kernel framework for hyperspectral image classification. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(9):4816–4829, 2013. 74, 94

[86] H. Liang and Q. Li. Hyperspectral imagery classification using sparse representations of convolutional neural network features. *Remote Sensing*, 8(2):99, 2016. 74

[87] J. Liang, J. Zhou, Y. Qian, L. Wen, X. Bai, and Y. Gao. On the sampling strategy for evaluation of spectral-spatial methods in hyperspectral image classification. *arXiv preprint arXiv:1605.05829*, 2016. 95

[88] W. Liao, X. Huang, F. Van Coillie, S. Gautama, A. Pižurica, W. Philips, H. Liu, T. Zhu, M. Shimoni, G. Moser, et al. Processing of multiresolution thermal hyperspectral and digital color data: Outcome of the 2014 ieee grss data fusion contest. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2984–2996, 2015. 117

[89] S. Limandri, G. Bernardi, and S. Suarez. Experimental study of the efficiency of a sdd x-ray detector by means of pixe spectra. *X-Ray Spectrometry*, 42(6):487–492, 2013. 112

[90] J. Liu. Smoothing filter-based intensity modulation: a spectral preserve image fusion technique for improving spatial details. *International Journal of Remote Sensing*, 21(18):3461–3472, 2000. 116

[91] L. Loncan, L. B. de Almeida, J. M. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. A. Licciardi, M. Simoes, et al. Hyperspectral pansharpening: a review. *IEEE Geoscience and remote sensing magazine*, 3(3):27–46, 2015. 110, 111, 113, 115, 124

[92] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 83

[93] X. Ma, J. Geng, and H. Wang. Hyperspectral image classification via contextual deep learning. *EURASIP Journal on Image and Video Processing*, 2015(1):1–12, 2015. 74

[94] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *Advances in Neural Information Processing Systems*, pages 2627–2635, 2014. 84

[95] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009. 30

[96] S. Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012. 7, 74, 76, 81, 82, 92

[97] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989. 117

[98] D. Manolakis, D. Marden, and G. A. Shaw. Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, 14(1):79–116, 2003. 13

[99] P. Maragos. Pattern spectrum and multiscale shape representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):701–716, 1989. 46

[100] G. Matheron. *Traité de géostatistique appliquée. 1 (1962)*, volume 1. Editions Technip, 1962. 74

[101] G. Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963. 140, 149

[102] G. Matheron. The intrinsic random functions and their applications. *Advances in applied probability*, pages 439–468, 1973. 140

[103] G. Matheron. *Random sets and integral geometry*. John Wiley & Sons, 1975. 47

[104] F. Mathieu, C. Jocelyn, B. Jón Atli, et al. Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. *EURASIP Journal on Advances in Signal Processing*, 2009, 2009. 62, 74

[105] F. Meier, P. Hennig, and S. Schaal. Incremental local gaussian regression. In *Advances in Neural Information Processing Systems*, pages 972–980, 2014. 154

[106] F. Meyer. The levelings. *COMPUTATIONAL IMAGING AND VISION*, 12:199–206, 1998. 60

[107] A. Mohan, G. Sapiro, and E. Bosch. Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images. *Geoscience and Remote Sensing Letters, IEEE*, 4(2):206–210, 2007. 62

[108] I. S. Molchanov and P. Teran. Distance transforms for real-valued functions. *Journal of Mathematical Analysis and Applications*, 278(2):472 – 484, 2003. 47, 54

[109] K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf. Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems*, pages 10–18, 2012. 74, 90

[110] K. Muandet, B. Sriperumbudur, K. Fukumizu, A. Gretton, and B. Schölkopf. Kernel mean shrinkage estimators. *Journal of Machine Learning Research*, 17(48):1–41, 2016. 85

[111] K. Muandet, B. Sriperumbudur, and B. Schölkopf. Kernel mean estimation via spectral filtering. In *Advances in Neural Information Processing Systems*, pages 1–9, 2014. 85

[112] D. Nguyen-Tuong, M. Seeger, and J. Peters. Model learning with local gaussian process regression. *Advanced Robotics*, 23(15):2015–2034, 2009. 154

[113] J. B. Oliva, D. J. Sutherland, B. Póczos, and J. Schneider. Deep mean maps. *arXiv preprint arXiv:1511.04150*, 2015. 74

[114] I. Olkin and A. W. Marshall. *Inequalities: theory of majorization and its applications*, volume 143. Academic press, 2016. 119

[115] G. K. Ouzounis, M. Pesaresi, and P. Soille. Differential area profiles: Decomposition properties and efficient computation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.34*, pages 1533–1548, Aug. 2012. 7

[116] K. Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901. 115

[117] M. Perrin. Principe du meb et microanalyse, centre de microscopie électronique à balayage et microanalyse, université de rennes i., `http://www.cmeba.univ-rennes1.fr/Principe_MEB.html`. 103, 104, 106

[118] M. Pesaresi and J. Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(2):309–320, Feb 2001. 74, 80, 94

[119] G. Piella. A general framework for multiresolution image fusion: from pixels to regions. *Information fusion*, 4(4):259–280, 2003. 110

[120] I. K. Qian Du and H. Szu. Article: Independent-component analysis for hyperspectral remote sensing imagery classification. *Optical Engineering*, Janvier 2006. 6

[121] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2007. 84

[122] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9(Nov):2491–2521, 2008. 25, 93

[123] T. Ranchin and L. Wald. Fusion of high spatial and spectral resolution images: the arsis concept and its implementation. *Photogrammetric Engineering and Remote Sensing*, 66(1):49–61, 2000. 116

[124] C. E. Rasmussen. Gaussian processes for machine learning. 2006. 147

[125] L. Reimer. *Scanning Electron Microscopy: Physics of Image Formation and Microanalysis*. Springer Verlag, Heidelberg, second edition, 1998. 111

[126] M. Rosenblatt et al. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956. 142

[127] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 6, 24

[128] F. Salvat, J. M. Fernández-Varea, and J. Sempau. Penelope-2006: A code system for monte carlo simulation of electron and photon transport. In *Workshop proceedings*, volume 7, 2006. 112

[129] S. S. Schiffman, F. W. Young, and M. L. Reynolds. *Introduction to multidimensional scaling: Theory, methods, and applications*. 1981. 17

[130] P. Schmid-Saugeon and A. Zakhor. Dictionary design for matching pursuit and application to motion-compensated video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(6):880–886, 2004. 30

[131] B. Schölkopf, C. Burges, and V. Vapnik. Incorporating invariances in support vector learning machines. In *International Conference on Artificial Neural Networks*, pages 47–52. Springer, 1996. 79

[132] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997. 6

[133] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002. 25, 32

[134] F. Scholze and M. Procop. Modelling the response function of energy dispersive x-ray spectrometers with silicon detectors. *X-Ray Spectrometry*, 38(4):312–321, 2009. 111

[135] J. Sempau, E. Acosta, J. Baro, J. Fernández-Varea, and F. Salvat. An algorithm for monte carlo simulation of coupled electron-photon transport. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 132(3):377–390, 1997. 112

[136] J. Sempau, J. Fernandez-Varea, E. Acosta, and F. Salvat. Experimental benchmarks of the monte carlo code penelope. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 207(2):107–123, 2003. 112

[137] J. Serra. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA, 1983. 45, 47, 79, 80

[138] C. Shalizi. *Advanced data analysis from an elementary point of view*. Citeseer, 2013. 143, 145

[139] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1233–1240. IEEE, 2013. 74

[140] A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *ALT*, pages 13–31, 2007. 7, 74, 83, 87, 91

Résumé

Cette thèse porte sur la statistique spatiale multivariée et l'apprentissage appliqués aux images hyperspectrales et multimodales. Les thèmes suivants sont abordés :

Fusion d'images :
Le microscope électronique à balayage (MEB) permet d'acquérir des images à partir d'un échantillon donné en utilisant différentes modalités. Le but de ces études est d'analyser l'intérêt de la fusion de l'information pour améliorer les images acquises par MEB. Nous avons mis en œuvre différentes techniques de fusion de l'information des images, basées en particulier sur la théorie de la régression spatiale. Ces solutions ont été testées sur quelques jeux de données réelles et simulées.

Classification spatiale des pixels d'images multivariées :
Nous avons proposé une nouvelle approche pour la classification de pixels d'images multi/hyper-spectrales. Le but de cette technique est de représenter et de décrire de façon efficace les caractéristiques spatiales / spectrales de ces images. Ces descripteurs multi-échelle profond visent à représenter le contenu de l'image tout en tenant compte des invariances liées à la texture et à ses transformations géométriques.

Réduction spatiale de dimensionnalité :
Nous proposons une technique pour extraire l'espace des fonctions en utilisant l'analyse en composante morphologiques. Ainsi, pour ajouter de l'information spatiale et structurelle, nous avons utilisé les opérateurs de morphologie mathématique.

Mots Clés

Traitement de l'image, Machine Learning, Méthodes à noyaux, Morphologie mathématique, Analyse en Composantes Principales, Support Vector Machine, Apprentissage profond, Transformée de scattering, Krigeage.

Abstract

This thesis focuses on multivariate spatial statistics and machine learning applied to hyperspectral and multimodal and images in remote sensing and scanning electron microscopy (SEM). In this thesis the following topics are considered:

Fusion of images:
SEM allows us to acquire images from a given sample using different modalities. The purpose of these studies is to analyze the interest of fusion of information to improve the multimodal SEM images acquisition. We have modeled and implemented various techniques of image fusion of information, based in particular on spatial regression theory. They have been assessed on various datasets.

Spatial classification of multivariate image pixels:
We have proposed a novel approach for pixel classification in multi/hyper-spectral images. The aim of this technique is to represent and efficiently describe the spatial/spectral features of multivariate images. These multi-scale deep descriptors aim at representing the content of the image while considering invariances related to the texture and to its geometric transformations.

Spatial dimensionality reduction:
We have developed a technique to extract a feature space using morphological principal component analysis. Indeed, in order to take into account the spatial and structural information we used mathematical morphology operators

Keywords

Image processing, Machine learning, Kernel methods, Mathematical morphology, Principal Component Analysis, Support Vector Machine, Deep learning, Scattering transform, Kriging