# Nonlinear unmixing of hyperspectral data with vector-valued kernel functions

Rita Ammanouil, André Ferrari, Member, IEEE, Cédric Richard, Senior Member, IEEE, Sandrine Mathieu

***Abstract*—This paper presents a kernel based nonlinear mixing model for hyperspectral data where the nonlinear function belongs to a Hilbert space of vector valued functions. The proposed model extends existing ones by accounting for band dependent and neighboring nonlinear contributions. The key idea is to work under the assumption that nonlinear contributions are dominant in some parts of the spectrum while they are less pronounced in other parts. In addition to this, we motivate the need for taking into account nonlinear contributions originating from the ground covers of neighboring pixels by practical considerations, precisely the adjacency effect. The relevance of the proposed model is that the nonlinear function is associated to a matrix valued kernel that allows to jointly model a wide range of nonlinearities and include prior information regarding band dependencies. Furthermore, the choice of the nonlinear function input allows to incorporate neighboring effects. The optimization problem is strictly convex and the corresponding iterative algorithm is based on the alternating direction method of multipliers (ADMM). Finally, experiments conducted using synthetic and real data demonstrate the effectiveness of the proposed approach.**

***Index Terms*—Hyperspectral imaging, nonlinear mixing, vector-valued RKHS, ADMM.**

## I. Introduction

Hyperspectral images are a very powerful tool in remote sensing. They provide a spectrum for each pixel, which is a vector of reflectance values estimated over hundreds of contiguous spectral bands with high spectral resolution [1], [2]. The spatial resolution of hyperspectral images can vary from a few meters up to a hundreds of meters in the case of airborne remote sensing. No matter the spatial resolution in this range, the image possibly contains mixed pixels where the surface covered by the pixel contains more than one constituent material [3]. This is in contrast with a pure pixel where the corresponding surface only contains one constituent material also known as an endmember. According to the well-known linear mixing model (LMM), the spectrum associated with a mixed pixel is a linear combination of the endmembers spectra [4]–[6]. More formally:

$$\boldsymbol{s}_n = \textstyle\sum_{i=1}^{M} a_{i,n}\boldsymbol{r}_i + \boldsymbol{e}_n, \forall\, n = 1, \dots N, \tag{1}$$

where $\boldsymbol{s}_n$ is the $L$-dimensional spectrum of the $n$-th pixel, $L$ is the number of frequency bands, $M$ denotes the number of endmembers, $a_{i,n}$ is the abundance of the $i$-th endmember in the $n$-th pixel, $\boldsymbol{r}_i$ is the $L$-dimensional spectrum of the $i$-th endmember, $\boldsymbol{e}_n$ is a vector of Gaussian white noise, and N is the number of observations. All vectors are column vectors. The abundances, which represent the relative contributions of the endmembers [7], are positive and usually sum to one: $a_{i,n} \geq 0$ and $\sum_{i=1}^{M} a_{i,n} = 1$ respectively. The LMM (1) is a simplified spectral mixing model. It only considers light reaching the sensor that has interacted once with the imaged surface, and neglects complex interactions between light, the imaged surface, neighboring surfaces and the atmosphere.

More recently, there has been a considerable amount of studies devoted to nonlinear mixing models [8], [9]. In particular, bilinear models are among the most widely known to account for nonlinear effects [10]–[13]. The physical assumption underlying these models is that light beams go through multiple reflections before reaching the sensor, mainly due to the three dimensionality of real scenes and scattering in the atmosphere [14]–[16]. The mathematical expression established for multiple reflections is the term by term product of two reflectance vectors in the case of bilinear models, and more than two reflectance vectors in the case of multilinear models [17]. For example, the polynomial post nonlinear mixing model (PPNM) introduced in [14] is a bilinear model that considers bilinear contributions through the following formulation:

$$\boldsymbol{s}_n = \sum_{i=1}^{M} a_{i,n}\boldsymbol{r}_i + u_n\Big(\sum_{i=1}^{M} a_{i,n}\boldsymbol{r}_i\Big) \odot \Big(\sum_{i=1}^{M} a_{i,n}\boldsymbol{r}_i\Big) + \boldsymbol{e}_n, \tag{2}$$

where $u_n$ is the nonlinearity parameter, and $\odot$ denotes the Hadamard (element wise) product. On the right hand side of equation (2), the first term corresponds to the linear mixture and the second term corresponds to the nonlinear (bilinear) one. Another way for modelling the nonlinear mixtures of the endmembers is through the use of nonlinear (scalar-valued) functions in reproducing kernel Hilbert spaces (RKHS) [18], [19]. The advantage of kernel-based models in RKHS over models similar to (2) is that they are non parametric which means that they do not impose a predetermined form for the nonlinearity. This provides a powerful and principled way for modeling a wide range of nonlinearities. It decouples the design of the mixing model and the corresponding optimization problem from the nonlinear mapping used for the nonlinear contribution. In fact, the kernel trick, which is the central idea in RKHS, allows to match the structure of the data through a nonlinear mapping without having to explicitly compute the mapping. See for example [18]–[20] for a review of the properties and definitions related to kernel-based methods in scalar RKHS. For example, the authors of [21] propose the

Rita Ammanouil, André Ferrari and Cédric Richard are with the Université Côte d'Azur, OCA, CNRS, Lagrange, France (e-mail: {rita.ammanouil, andre.ferrari, cedric.richard}@unice.fr). Sandrine Mathieu is with Thales Alenia Space, Cannes la Bocca, France (e-mail: sandrine.mathieu@thalesaleniaspace.com). This work was partly supported by the Agence Nationale pour la Recherche, France, (Hypanema project, ANR-12-BS03-003), and the regional council of Provence Alpes-Côte d'Azur. The authors would like to thank the European Space Agency for providing the Meris data set.

following model:

$$s_n \quad = \sum_{i=1}^{M} a_{i,n} \boldsymbol{r}_i + \boldsymbol{\Psi}_n(\boldsymbol{R}) + \boldsymbol{e}_n, \tag{3}$$

where $\boldsymbol{\Psi}_n(\boldsymbol{R}) = [\Psi_n(\boldsymbol{r}_{\lambda_1}) \dots \Psi_n(\boldsymbol{r}_{\lambda_L})]^\top$, $\Psi_n(\cdot)$ is a nonlinear function in a reproducing kernel Hilbert space (RKHS), and $\boldsymbol{r}_{\lambda_i}$ denotes the $i$-th row in $\boldsymbol{R}$. The authors of [21] show that model (3) is able to incorporate bilinear, multilinear as well as more complex nonlinear interactions between the endmembers depending on the kernel choice and its associated mapping. The main drawback of both, bilinear and (scalar-valued) kernel based models, is that they impose the same function at all bands, which can be restrictive in practice. More precisely, bilinear models consider the same amount of bilinear contributions at all bands. In the case of the PPNM (2), the same weight $u_n a_{i,n} a_{j,n}$ is used to scale the bilinear contribution of $\boldsymbol{r}_i \odot \boldsymbol{r}_j$ at all bands. Similarly, the kernel-based model (3) considers the same scalar-valued nonlinear function $\Psi_n(\cdot)$ at all bands. In contrast with the aforementioned models, the authors of [22], [23] do not impose any analytical form for the nonlinear term. The nonlinear contribution is merely treated as a positive residual term that is sparsely (rarely) present among the observations [23]. Nevertheless, this model-free approach can be limiting itself since it does not control the nonlinear expression, hence it can prevent accurate estimations of the nonlinear contribution.

The proposed nonlinear mixing model circumvents the drawbacks of the previously cited models by assuming that the nonlinear function belongs to a reproducing kernel Hilbert space (RKHS) of vector-valued functions. This approach improves upon the case of RKHS of scalar-valued functions by allowing for variable nonlinear contributions at different bands. The key idea is to work under the assumption that nonlinear contributions can be dominant in some parts of the spectrum and less significant in other parts [24]–[26]. Unlike the scalar valued case where the same function is considered at each band, the proposed model relaxes this constraint, and allows to take into account wavelength dependent nonlinear contributions. In particular, we focus on RKHS associated with a special type of kernels, namely separable kernels. This type of kernels jointly defines the form of the nonlinear contribution, and allows to include prior information regarding the similarity between the nonlinear contributions at different wavelength bands. Similarly to the PPNM model, the input of the nonlinear vector-valued function includes the linear mixture spectra present in the pixel. We go one step further by also including the spectra of the linear mixtures in neighboring pixels. This choice is motivated by the adjacency effect [25] which states that solar radiation reflected off adjacent surfaces can be scattered into the sensor's instantaneous field of view. Figure 1 shows two forms of the adjacency effect, and depicts how neighboring surfaces can nonlinearly contribute to the reflectance vector estimated for a pixel. The adjacency effects are usually removed in a pre-processing step known as atmospheric correction. There exist different empirical methods for atmospheric correction [25]–[27]. Nevertheless, the validity of some of these methods to correct the adjacency effect is still questionnable [28], and errors occurred by these methods can damage the quality of information extracted from remote sensing data [29]. As a result, accounting for potential adjacency effects through the input of the nonlinear function increases the mixing model accuracy.

The paper is organized as follows. Section II describes the nonlinear mixing model, and discusses approaches for constructing the matrix valued kernel, section III develops the optimization problem and the corresponding estimation algorithm. Finally, section IV validates the proposed mixing model using synthetic and real data.

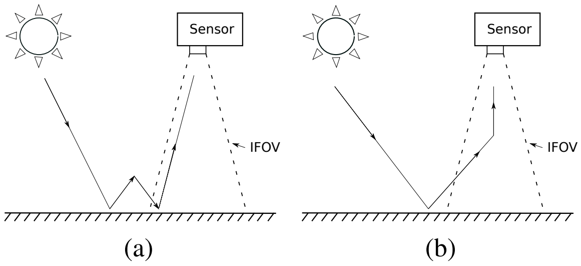


Fig. 1. Illustration of two forms of the adjacency effect resulting from: (a) multiple reflections involving the targeted surface and an adjacent surface, (b) reflection off an adjacent surface that is then scattered in the atmosphere into the sensor's instantaneous field of view (IFOV).

## II. Nonlinear mixing model

### A. Model Description

First, we assume that the image is partitioned into patches or groups of pixels, and that the pixels in each patch are associated with a vector-valued nonlinear function. For ease of notations, assume that the available observations $\boldsymbol{S} = [\boldsymbol{s}_1, \dots, \boldsymbol{s}_N]$ belong to the same patch and that they are associated with the function $\boldsymbol{f}$. The proposed nonlinear model decomposes the spectrum of a pixel into the sum of a linear and nonlinear part:

$$\boldsymbol{s}_n \quad = \boldsymbol{s}_n^{\text{lin}} + \boldsymbol{f}(\tilde{\boldsymbol{v}}_n) + \boldsymbol{e}_n, \tag{4}$$

where $\boldsymbol{s}_n^{\text{lin}} = \sum_{i=1}^{M} a_{i,n} \boldsymbol{r}_i$, $\tilde{\boldsymbol{v}}_n = \text{col}(\{\boldsymbol{s}_i^{\text{lin}}\}_{i \in \mathcal{C}_n})$, $\text{col}(\cdot)$ is an operator that stacks its arguments on top of each other, and $\mathcal{C}_n$ is a set of indices of $c$ pixels including the $n$-th pixel and its $c-1$ neighbours (for example $\mathcal{C}_n = \{n, n-1, n+1\}$). The nonlinear contribution in (4) is expressed in terms of the pixel and its neighbors linear mixtures. This is in accordance with several bilinear models such as the post polynomial nonlinear mixing (PPNM) model [13] which expresses the nonlinear contribution solely in terms of $\boldsymbol{s}_n^{\text{lin}}$. The inconvenience of model (4) is that $\tilde{\boldsymbol{v}}_n$ depends on the unknown abundances. As a result, the corresponding optimization problem is not convex since the function $\boldsymbol{f}$ is unknown itself. In order to have a convex optimization problem, we propose to approximate $\boldsymbol{s}_i^{\text{lin}}$ by $\boldsymbol{s}_i$, hence equation (4) becomes:

$$\boldsymbol{s}_n \quad = \boldsymbol{s}_n^{\text{lin}} + \boldsymbol{f}(\boldsymbol{v}_n) + \boldsymbol{e}_n, \tag{5}$$

where $\boldsymbol{v}_n = \text{col}(\{\boldsymbol{s}_i\}_{i \in \mathcal{C}_n})$, and it is assumed that $\boldsymbol{f}(\boldsymbol{v}_n)$ and $\boldsymbol{e}_n$ are small compared to $\boldsymbol{s}_n^{\text{lin}}$. The latter assumption holds provided that the signal to noise ratio (SNR) is relatively high and that the linear part dominates the nonlinear one. The vector-valued approach offers an elegant and flexible way to jointly estimate multiple nonlinear functions at all bands since $\boldsymbol{f}(\boldsymbol{v}_n)$ implicitly corresponds to having different scalar-valued functions per band, i.e. $\boldsymbol{f}(\boldsymbol{v}_n) = [f_1(\boldsymbol{v}_n) \dots f_L(\boldsymbol{v}_n)]^\top$. As mentioned previously, the same nonlinear function is associated with all the pixels in the corresponding patch. It is important to note that the patch should contain enough pixels in order to have a good estimation of the nonlinear function. Moreover, it should be small enough to reflect the variability of the nonlinear function in the different regions of the image.

### *B. RKHS of vector-valued functions*

The nonlinear function $\boldsymbol{f}$ used in (5) is a vector-valued function, its evaluation is a vector with $L$ components representing the nonlinear contributions present in $\boldsymbol{s}_n$ at each band:

$$\begin{array}{rccl} \boldsymbol{f}: & \mathbb{R}^{L\,c} & \to & \mathbb{R}^L \\ & \boldsymbol{v}_n & \to & \boldsymbol{f}(\boldsymbol{v}_n). \end{array} \tag{6}$$

As mentioned previously, we assume that $\boldsymbol{f}$ belongs to $\widetilde{\mathcal{H}}_{\boldsymbol{k}}$, a RKHS of vector-valued functions associated with the following kernel function:

$$\begin{array}{rccl} \widetilde{\boldsymbol{k}}: & \mathbb{R}^{L\,c} \times \mathbb{R}^{L\,c} & \to & \mathbb{R}^{L\times L} \\ & (\boldsymbol{v}_n, \boldsymbol{v}_{n'}) & \to & \widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'}). \end{array} \tag{7}$$

Unlike the scalar-valued case, the kernel is a positive semi-definite matrix in $\mathbb{R}^{L\times L}$ rather than a positive scalar value. The overall Gram matrix $\widetilde{\boldsymbol{K}}$ associated with the function $\boldsymbol{f}$ is the matrix obtained from the evaluation of the kernel function (7) at all observation couples. It is a block matrix, such that the block indexed by $(n, n')$ is given by:

$$\widetilde{\boldsymbol{K}}_{n,n'} = \widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'}). \tag{8}$$

The Gram matrix consists of $N \times N$ blocks, where each block is an $L \times L$ matrix as defined in (8). As a result, $\widetilde{\boldsymbol{K}}$ is an $LN \times LN$ matrix. The representer theorem for vector-valued functions parallels the theorem in the scalar valued case. According to [30], $\boldsymbol{f}(\boldsymbol{v}_n)$ can be expressed as an expansion of the kernel function over all training points:

$$\boldsymbol{f}(\boldsymbol{v}_n) = \textstyle\sum_{n'=1}^{N} \widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'})\boldsymbol{\alpha}_{n'}, \tag{9}$$

where $\boldsymbol{\alpha}_{n'} \in \mathbb{R}^L$. As a result, estimating the nonlinear function reduces to estimating the coefficients $\{\boldsymbol{\alpha}_{n'}\}_{n'=1}^{N}$. As mentioned previously, if $\tilde{\boldsymbol{v}}_n$ was not approximated with $\boldsymbol{v}_n$ the resulting optimization problem would be non-convex. It can be seen from equation (9) that the expression of $\boldsymbol{f}(\tilde{\boldsymbol{v}}_n)$ incorporates a product between unknown variables, namely between the kernel function acting on the unknown abundances (through $\tilde{\boldsymbol{v}}_n$) and the unknown kernel coefficients. Finally, the norm of the function $\boldsymbol{f}$ in the RKHS $\widetilde{\mathcal{H}}_{\boldsymbol{k}}$ can be written as:

$$\|\boldsymbol{f}\|^2_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}} = \sum_{n=1}^{N}\sum_{n'=1}^{N} \boldsymbol{\alpha}_n^\top \widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'})\boldsymbol{\alpha}_{n'}, \tag{10}$$

which gives a natural measure of the complexity of the function [30], [31].

### *C. Kernel design*

The kernel function $\widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'})$, as defined in equation (7), is an $L \times L$ matrix. The design of the kernel allows to jointly define the nonlinearity and include prior information regarding the similarity between the outputs of the nonlinear function at different bands. The authors of [30], [32] describe two possible kernel classes known as the transformable and the separable kernels. In what follows we describe these two classes of kernels, and explain their relevance in the nonlinear unmixing context.

*1) Transformable and separable kernels:* The first class of matrix valued kernels is known as transformable kernels. In the transformable case, the kernel $\widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'})$ is defined in a component-wise fashion through a scalar valued kernel. Each component of the kernel is defined as:

$$\Big[\widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'})\Big]_{\ell,\ell'} = k(\boldsymbol{T}_\ell \boldsymbol{v}_n, \boldsymbol{T}_{\ell'} \boldsymbol{v}_{n'}), \tag{11}$$

where $k$ is a scalar valued kernel, and $\boldsymbol{T}_\ell$ is an operator that extracts the $c$ reflectance values in $\boldsymbol{v}_n$ corresponding to the $\ell$-th band. More precisely, $\boldsymbol{T}_\ell \boldsymbol{v}_n = \mathrm{col}(\{s_{\ell,i}\}_{i\in\mathcal{C}_n})$ is a vector with $c$ components. The scalar valued kernel acts jointly on pixels and bands indices, $(n, n')$ and $(\ell, \ell')$ respectively. The relevance of the transformable kernel is that the Gram matrix encodes the similarity between all pairs of pixels at all bands. Hence, it exploits all these correlations in order to jointly estimate the $L$ components of the nonlinear function.

The second class of matrix valued kernels is known as the separable kernels. This class of kernels allows to incorporate prior information regarding the similarities between the different components of the vector valued function. The separable kernel is defined as the product between a scalar kernel acting on the input and an $L \times L$ positive semi-definite matrix encoding the similarities between the nonlinear contributions at different bands, i.e. between the different components of $\boldsymbol{f}$. For this class, the kernel is defined as follows:

$$\widetilde{\boldsymbol{k}}(\boldsymbol{v}_n, \boldsymbol{v}_{n'}) = k(\boldsymbol{v}_n, \boldsymbol{v}_{n'})\boldsymbol{\mathcal{E}}, \tag{12}$$

where $\boldsymbol{\mathcal{E}}$ is an $L \times L$ positive semi-definite matrix. The norm of $\boldsymbol{f}$ gives further insight on how $\boldsymbol{\mathcal{E}}$ encodes the similarities between the nonlinear contributions at different bands. In fact, the norm of $\boldsymbol{f}$ in $\widetilde{\mathcal{H}}_{\boldsymbol{k}}$ [33], [34] is given by:

$$\|\boldsymbol{f}\|^2_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}} = \sum_{\ell=1}^{N}\sum_{\ell'=1}^{N} \mathcal{E}^\dagger_{\ell,\ell'} \langle f_\ell, f_{\ell'} \rangle_{\mathcal{H}_k}, \tag{13}$$

where $\boldsymbol{\mathcal{E}}^\dagger$ is the pseudo inverse of $\boldsymbol{\mathcal{E}}$, the scalar valued nonlinear functions $f_1, \dots, f_L$ belong to the RKHS $\mathcal{H}_k$ associated with $k$, such that $\boldsymbol{f} = [f_1, \dots, f_L]^\top$. Note that in the case of the separable kernel, given (8) and (12), the overall Gram matrix can be written in the following form:

$$\widetilde{\boldsymbol{K}} = \boldsymbol{K} \otimes \boldsymbol{\mathcal{E}}, \tag{14}$$

where $\otimes$ is the Kronecker product, and $\boldsymbol{K}$ is the $N \times N$ Gram matrix associated with the scalar valued kernel, namely $k_{n,n'} = k(\boldsymbol{v}_n, \boldsymbol{v}_{n'})$. Hereafter, we investigate a special structure for the matrix $\boldsymbol{\mathcal{E}}$. First, we assume that there exits prior information about the closeness between nonlinear contributions at different bands, i.e. between the functions $f_1, \ldots, f_L$. This prior can be modeled by a graph. We denote by $\boldsymbol{W} \in \mathbb{R}^{L \times L}$ the adjacency matrix of this graph [35]. More precisely, when two bands are likely to have similar nonlinear contributions, the corresponding nodes are connected by an edge and associated with a positive similarity weight $w_{\ell,\ell'} > 0$, otherwise $w_{\ell,\ell'}$ is set to zero. The authors of [33] show that when $\boldsymbol{\mathcal{E}}^{\dagger}$ is related to $\boldsymbol{W}$ as follows:

$$\begin{cases} \mathcal{E}^{\dagger}_{\ell,\ell'} & = -w_{\ell,\ell'}, \text{ if } \ell \neq \ell', \\ \mathcal{E}^{\dagger}_{\ell,\ell} & = \sum_{\ell'=1}^{N} w_{\ell,\ell'}, \text{ otherwise,} \end{cases} \tag{15}$$

using (13), the norm of $\boldsymbol{f}$ in $\widetilde{\mathcal{H}}_{\boldsymbol{k}}$ can be rewritten as:

$$\|\boldsymbol{f}\|^2_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}} = \sum_{\ell=1}^{L} \|f_\ell\|^2_{\mathcal{H}_k} w_{\ell,\ell} + \frac{1}{2} \sum_{\ell=1}^{L} \sum_{\ell'=1}^{L} \|f_\ell - f_{\ell'}\|^2_{\mathcal{H}_k} w_{\ell,\ell'}. \tag{16}$$

From a regularization point of view, the norm of $\boldsymbol{f}$ as given by equation (16) is known as the graph regularizer. It penalizes the norms of the individual functions in addition to the differences between each pair of functions, hence forcing them to be similar. Moreover, the strength of the similarity between each pair of functions is determined by the corresponding weight. More precisely, a high value of $w_{\ell,\ell'}$ promotes a strong similarity between $f_\ell$ and $f_{\ell'}$, and conversely, a low value of $w_{\ell,\ell'}$ promotes a weak similarity between the two functions. In other words, the norm of $\boldsymbol{f}$ as given by equation (16) promotes similarity between the estimated nonlinearities at different bands in accordance with the prior information reflected through the graph. Finally, note that when $\boldsymbol{\mathcal{E}} = \boldsymbol{I}_L$, the norm of $\boldsymbol{f}$ reduces to the sum of the individual norms of its components $f_{\ell'}$. This corresponds to the case where there is no prior information between the nonlinearities at different bands.

In general, it is very likely that the nonlinear contributions at consecutive bands have smooth spectral variations. This prior information can be represented by a linear graph, where each node is connected to nodes at adjacent bands with unit weight. Nevertheless, more complex similarities can be incorporated using the graph structure. For example, in certain scenes nonlinear contributions can be dominant in certain spectral domains and less dominant in other spectral domains [24], [36]. This prior information can be represented by a clustered graph, where only nodes in spectral domains with similar nonlinear behavior are connected to each other.

*2) Scalar kernel choice:* The transformable and the separable kernels are both defined respectively in equations (11) and (12) using a scalar valued kernel $k$. Similarly to the case of functions in scalar valued RKHS, the choice of the kernel corresponds to a certain representation of the input data in a higher dimensional space known as the feature space [19]. Hence, looking at the feature space can provide guidelines for choosing an appropriate kernel.

We focus on the polynomial and Gaussian kernels due to their successful application to nonlinear unmixing in the scalar valued case [21], [37]. In particular, the second order homogeneous polynomial kernel:

$$k(\boldsymbol{v}_n, \boldsymbol{v}_{n'}) = (\boldsymbol{v}_n{}^\top \boldsymbol{v}_{n'})^2, \tag{17}$$

can be written as the inner product of the feature maps of $\boldsymbol{v}_n$ and $\boldsymbol{v}_{n'}$, where the feature map is defined as follows:

$$\begin{aligned} \phi(\boldsymbol{v}_n) = [&(v_{n,1})^2, \ldots (v_{n,Lc})^2, \sqrt{2}(v_{n,1}v_{n,2})^2, \\ &\ldots \sqrt{2}(v_{n,1}v_{n,Lc})^2, \sqrt{2}(v_{n,2}v_{n,3})^2, \ldots \\ &\sqrt{2}(v_{n,2}v_{n,Lc})^2, \ldots \sqrt{2}(v_{n,Lc-1}v_{n,Lc})^2]. \end{aligned} \tag{18}$$

The feature map of the second order homogeneous polynomial kernel maps its input vector to all the possible pairwise products between its components. This can be seen as a representation of all possible second order interactions between the spectral values in the input vector. On the other hand, the Gaussian kernel:

$$k(\boldsymbol{v}_n, \boldsymbol{v}_{n'}) = \exp(-\frac{\|\boldsymbol{v}_n - \boldsymbol{v}_{n'}\|^2}{2\sigma^2}), \tag{19}$$

can be expressed as an infinite series of higher order polynomial kernels:

$$k(\boldsymbol{v}_n, \boldsymbol{v}_{n'}) = \sum_{j=0}^{\infty} \frac{(\boldsymbol{v}_n{}^\top \boldsymbol{v}_{n'})^j}{\sigma^{2j} j!} \exp(-\frac{\|\boldsymbol{v}_n\|^2}{2\sigma^2}) \exp(-\frac{\|\boldsymbol{v}_{n'}\|^2}{2\sigma^2}). \tag{20}$$

Theoretically, the Gaussian kernel represents the case where an endless number of reflections occurs in the scene since it incorporates all higher order interactions between the input spectra. The drawback of the Gaussian kernel is that its feature map also contains a constant and a linear contribution (for $j = 0$ and $j = 1$ in equation (20)) which can hinder the estimation accuracy. Nevertheless, the Gaussian kernel shows satisfying results in practice as will be seen in the experiments.

## III. Estimation algorithm

### *A. Optimization problem*

In this section we derive the optimization problem aimed at estimating the abundances and the nonlinear function based on model (5). Assuming that the noise is white, Gaussian, with zero mean, and a possibly unknown variance, adopting a maximum likelihood estimation leads to the least square (LS) optimization problem:

$$\min_{\boldsymbol{A}, \boldsymbol{f} \in \widetilde{\mathcal{H}}_{\boldsymbol{k}}} \quad \tfrac{1}{2} \textstyle\sum_{n=1}^{N} \|\boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{a}_n - \boldsymbol{f}(\boldsymbol{v}_n)\|^2, \tag{21}$$

where $\boldsymbol{R} = [\boldsymbol{r}_1, \ldots, \boldsymbol{r}_M]$, and $\boldsymbol{a}_n = [a_{1,n}, \ldots, a_{M,n}]^\top$. Problem (21) mainly ensures that the estimated model matches the observations. Nevertheless, the estimation of the abundances and the nonlinear function based on (21) is an underdetermined problem. As a result, it requires regularization and taking into account additional constraints on the abundances. For these reasons, we shall consider the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{A}, \boldsymbol{f} \in \widetilde{\mathcal{H}}_{\boldsymbol{k}}} \quad & \tfrac{1}{2} \textstyle\sum_{n=1}^{N} \|\boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{a}_n - \boldsymbol{f}(\boldsymbol{v}_n)\|^2 \\ & + \tfrac{\lambda}{2} \|\boldsymbol{f}\|^2_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}} + \mu \mathcal{J}(\boldsymbol{A}) \\ \text{subject to} \quad & a_{i,n} \succeq 0 \quad \forall\, i = 1, \cdots M, n = 1, \cdots N, \\ & \textstyle\sum_{i=1}^{M} a_{i,n} = 1 \quad \forall\, n = 1, \cdots, N, \end{aligned} \tag{22}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$, $\lambda$ and $\mu$ are tuning parameters that control the tradeoff between the LS term and the two regularization terms. The first regularization, namely the $\ell_2$-norm of $\boldsymbol{f}$ in $\widetilde{\mathcal{H}}_{\boldsymbol{k}}$, constrains the complexity of the estimated function [31]. Furthermore, in the case of the separable kernel, it corresponds to the graph regularizer (16) which promotes smoothness between the outputs of $\boldsymbol{f}$ according to the graph structure. As a result, the tuning parameter $\lambda$ controls the trade-off between fitting the data and smoothing the nonlinear contributions by scaling the graph weights in expression (16). If $\lambda$ is set to zero, then no smoothness is promoted between the spectral bands. On the other hand, the larger the value of $\lambda$, the smoother the nonlinear contributions at neighboring spectral bands in the graph. The second regularizer, namely $\mathcal{J}(\boldsymbol{A})$, aims at incorporating prior information about the abundances. For example, in the experiments we use the Frobenius norm of the abundances:

$$\mathcal{J}(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{A}\|_{\mathrm{F}}^2, \tag{23}$$

giving preference to abundance estimates having a small squared $\ell_2$ norm for each pixel. Note that for very large values of the tuning parameter $\mu$ in (22) the abundance estimates for all pixels and all endmembers tend to be equal to $\frac{1}{M}$ due to the positivity and sum-to-one constraints. As shown in the next section III-B, using another expression for $\mathcal{J}(\boldsymbol{A})$ is not cumbersome and affects one step in the iterative algorithm. Nevertheless, an advantage of using the $\ell_2$ norm of the nonlinear function and the Frobenius norm of the abundances is that each regularization is strictly convex with respect to the corresponding unknown variable. Hence, the overall optimization problem is strictly convex with respect to all the unknown variables. Finally, the proposed optimization problem (22) imposes the positivity and sum-to-one constraints on the estimated abundances. Some of the nonlinear mixing models in the literature keep the sum-to-one constraint, as for example [13], [37]. It can be argued that this constraint should be relaxed to $\sum_{i=1}^{M} a_{i,n} \leq 1$ especially when dealing with real hyperspectral data. Even if this constraint is strictly enforced in the proposed optimization problem (22), it can be relaxed by introducing a shade endmember in the endmember matrix [38]. Furthermore, we show in the next section that dropping this constraint requires a simple modification of the iterative algorithm.

### B. Iterative algorithm

In this section, we use the alternating direction method of multipliers (ADMM) [39] to solve the proposed optimization problem (22). The ADMM is a primal dual splitting method based on the augmented Lagrangian [40], [41]. Following the ADMM strategy, new variables and the corresponding consensus constraints are introduced in (22) in order to decouple the various terms in the objective function. We reformulate the optimization problem (22) in the following equivalent manner:

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{Z},\boldsymbol{f}\in\widetilde{\mathcal{H}}_{\boldsymbol{k}}} \quad & \tfrac{1}{2}\textstyle\sum_{n=1}^{N}\|\boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{x}_n - \boldsymbol{f}(\boldsymbol{v}_n)\|^2 \\ & +\tfrac{\lambda}{2}\|\boldsymbol{f}\|_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}}^2 + \mu\mathcal{J}(\boldsymbol{Z}) + \mathcal{I}_{\mathbb{R}_+^{M\times N}}(\boldsymbol{Z}) \\ \text{subject to} \quad & \boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} = \boldsymbol{\mathcal{C}}, \end{aligned} \tag{24}$$

with

$$\boldsymbol{\mathcal{A}} = \begin{pmatrix} \boldsymbol{I}_{M\times M} \\ \boldsymbol{1}_M^\top \end{pmatrix}, \; \boldsymbol{\mathcal{B}} = \begin{pmatrix} -\boldsymbol{I}_{M\times M} \\ \boldsymbol{0}_M^\top \end{pmatrix}, \; \boldsymbol{\mathcal{C}} = \begin{pmatrix} \boldsymbol{0}_{M\times N} \\ \boldsymbol{1}_N^\top \end{pmatrix} \tag{25}$$

where $\boldsymbol{X}$ and $\boldsymbol{Z}$ are the ADMM variables, and $\mathcal{I}_{\mathbb{R}_+^{M\times N}}(\boldsymbol{Z})$ is the indicator of $\mathbb{R}_+^{M\times N}$ (i.e., $\mathcal{I}_{\mathbb{R}_+^{M\times N}}(\boldsymbol{Z}) = 0$ if $\boldsymbol{Z} \in \mathbb{R}_+^{M\times N}$ and $\mathcal{I}_{\mathbb{R}_+^{M\times N}}(\boldsymbol{Z}) = \infty$ if $\boldsymbol{Z} \notin \mathbb{R}_+^{M\times N}$). Compared to problem (22), $\boldsymbol{A}$ was substituted by $\boldsymbol{X}$ and $\boldsymbol{Z}$, and a consensus constraint between the new variables was introduced. The positivity constraint was moved to the objective function through the indicator function, and the sum-to-one was incorporated within the equality constraint. As mentioned in the previous section, the sum-to-one constraint can be relaxed by adding a shade endmember to $\boldsymbol{R}$. Another alternative for relaxing the sum-to-one is by changing the definition of the matrices in (25) to:

$$\boldsymbol{\mathcal{A}} = \boldsymbol{I}_{M\times M}, \quad \boldsymbol{\mathcal{B}} = -\boldsymbol{I}_{M\times M}, \quad \boldsymbol{\mathcal{C}} = \boldsymbol{0}_{M\times N}. \tag{26}$$

The augmented Lagrangian associated with problem (24) is given by:

$$\begin{aligned} \mathcal{L}_\rho(\boldsymbol{X},\boldsymbol{Z},\boldsymbol{f},\boldsymbol{\Lambda}_\rho) = \tfrac{1}{2}\textstyle\sum_{n=1}^{N}\|\boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{x}_n - \boldsymbol{f}(\boldsymbol{v}_n)\|^2 \\ +\tfrac{\lambda}{2}\|\boldsymbol{f}\|_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}}^2 + \mu\mathcal{J}(\boldsymbol{Z}) + \mathcal{I}_{\mathbb{R}_+^{M\times N}}(\boldsymbol{Z}) \\ +\mathrm{tr}(\boldsymbol{\Lambda}_\rho^\top(\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}})) + \tfrac{\rho}{2}\,\|\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}\|_{\mathrm{F}}^2, \end{aligned} \tag{27}$$

where $\boldsymbol{\Lambda}_\rho$ is the matrix of Lagrange multipliers associated with the linear constraints in (24), and $\rho$ is the penalty parameter. At each iteration, the ADMM algorithm consists of minimizing the augmented Lagrangian (27) sequentially. First, it is minimized with respect to the unknown variables $\{\boldsymbol{X}, \boldsymbol{f}\}$ and then with respect to $\boldsymbol{Z}$ while in each minimization keeping the other variables fixed to their previous estimate. Finally, it consists of updating the Lagrange multipliers matrix $\boldsymbol{\Lambda}_\rho$ associated with the linear constraints. This approach allows to break the optimization problem into a sequence of smaller and simpler sub-problems. The ADMM steps at each iteration, namely the $\{\boldsymbol{X}, \boldsymbol{f}\}$ minimization step, the $\boldsymbol{Z}$ minimization step, and the update of the Lagrange multipliers, are developed hereafter. To keep the notations simple, we drop the iteration index.

*1)* $\{\boldsymbol{X}, \boldsymbol{f}\}$ **minimization step:** This step consists of minimizing the augmented Lagrangian with respect to $\{\boldsymbol{X}, \boldsymbol{f}\}$. After discarding the terms independent of $\{\boldsymbol{X}, \boldsymbol{f}\}$ in (27), this step reduces to the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{f}\in\widetilde{\mathcal{H}}_{\boldsymbol{k}}} \tfrac{1}{2}\sum_{n=1}^{N}\|\boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{x}_n - \boldsymbol{f}(\boldsymbol{v}_n)\|^2 + \tfrac{\lambda}{2}\|\boldsymbol{f}\|_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}}^2 + \\ \mathrm{tr}(\boldsymbol{\Lambda}_\rho^\top\boldsymbol{\mathcal{A}}\boldsymbol{X}) + \tfrac{\rho}{2}\,\|\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}\|_{\mathrm{F}}^2. \end{aligned} \tag{28}$$

We rewrite (28) in the following equivalent form:

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{f}\in\widetilde{\mathcal{H}}_{\boldsymbol{k}},\boldsymbol{E}} \quad & \tfrac{1}{2}\sum_{n=1}^{N}\|\boldsymbol{e}_n\|^2 + \tfrac{\lambda}{2}\|\boldsymbol{f}\|_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}}^2 + \mathrm{tr}(\boldsymbol{\Lambda}_\rho^\top\boldsymbol{\mathcal{A}}\boldsymbol{X}) \\ & +\tfrac{\rho}{2}\,\|\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \boldsymbol{e}_n = \boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{x}_n - \boldsymbol{f}(\boldsymbol{v}_n), \\ & \forall\, n = 1, \cdots, N, \end{aligned} \tag{29}$$

where $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N]$, and solve its dual problem. The Lagrangian associated with problem (29) is given by:

$$\begin{aligned}\mathcal{L}(\boldsymbol{X}, \boldsymbol{f}, \boldsymbol{E}, \boldsymbol{\Lambda}) = \tfrac{1}{2} \sum_{n=1}^{N} \|\boldsymbol{e}_n\|^2 + \tfrac{\lambda}{2} \|\boldsymbol{f}\|^2_{\widetilde{\mathcal{H}}_{\boldsymbol{k}}} + \operatorname{tr}(\boldsymbol{\Lambda}_{\rho}^{\top} \boldsymbol{\mathcal{A}} \boldsymbol{X}) \\ + \textstyle\sum_{n=1}^{N} \boldsymbol{\lambda}_n^{\top} (\boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{x}_n - \boldsymbol{f}(\boldsymbol{v}_n) - \boldsymbol{e}_n) \\ + \tfrac{\rho}{2} \, \|\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}\|^2_{\mathrm{F}},\end{aligned} \tag{30}$$

where $\boldsymbol{\Lambda} = [\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_N]$ is the matrix of Lagrange multipliers associated with the linear constraints in (29). The partial derivatives of the Lagrangian with respect to the primal variables, namely $\boldsymbol{X}, \boldsymbol{f}$ and $\boldsymbol{E}$, are:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \boldsymbol{X}} & = \rho \boldsymbol{\mathcal{A}}^{\top} \boldsymbol{\mathcal{A}} \boldsymbol{X} - \boldsymbol{R}^{\top} \boldsymbol{\Lambda} + \boldsymbol{\mathcal{A}}^{\top} \boldsymbol{\Lambda}_{\rho} + \rho \boldsymbol{\mathcal{A}}^{\top} (\boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}), \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{f}} & = \lambda \boldsymbol{f}(\cdot) - \frac{1}{\lambda} \sum_{j=1}^{N} \widetilde{\boldsymbol{k}}(\cdot, \boldsymbol{v}_j) \boldsymbol{\lambda}_j, \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{E}} & = \boldsymbol{E} - \boldsymbol{\Lambda}. \end{cases} \tag{31}$$

Setting the gradient of the partial derivatives in (31) to zero gives the primal variables as a function of the Lagrange multipliers:

$$\begin{cases} \boldsymbol{X}^* & = \frac{(\boldsymbol{\mathcal{A}}^{\top}\boldsymbol{\mathcal{A}})^{-1}}{\rho} (\boldsymbol{R}^{\top} \boldsymbol{\Lambda}^* - \boldsymbol{\mathcal{A}}^{\top} \boldsymbol{\Lambda}_{\rho} - \rho \boldsymbol{\mathcal{A}}^{\top} (\boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}})), \\ \boldsymbol{f}^*(\cdot) & = \frac{1}{\lambda} \sum_{j=1}^{N} \widetilde{\boldsymbol{k}}(\cdot, \boldsymbol{v}_j) \boldsymbol{\lambda}_j^*, \\ \boldsymbol{E}^* & = \boldsymbol{\Lambda}^*. \end{cases} \tag{32}$$

To derive the Lagrange dual problem, the primal variables are substituted in (30) by their expressions from (32). This results in a quadratic form with respect to the Lagrange multipliers, and yields the following dual problem:

$$\max_{\boldsymbol{\Lambda}} \quad -\tfrac{1}{2} \operatorname{vec}(\boldsymbol{\Lambda})^{\top} \boldsymbol{Q} \operatorname{vec}(\boldsymbol{\Lambda}) + \operatorname{vec}(\boldsymbol{\Lambda})^{\top} \boldsymbol{p}, \tag{33}$$

with

$$\begin{cases} \boldsymbol{Q} & = \boldsymbol{I}_{LN \times LN} + \frac{1}{\lambda} \widetilde{\boldsymbol{K}} + \frac{1}{\rho} \boldsymbol{I}_{N \times N} \otimes \boldsymbol{D}, \\ \boldsymbol{p} & = \operatorname{vec}(\boldsymbol{S} + \frac{1}{\rho} \boldsymbol{R} (\boldsymbol{\mathcal{A}}^{\top}\boldsymbol{\mathcal{A}})^{-1} \boldsymbol{\mathcal{A}}^{\top} (\boldsymbol{\Lambda}_{\rho} + \rho(\boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}))), \end{cases} \tag{34}$$

where $\boldsymbol{D} = \boldsymbol{R}(\boldsymbol{\mathcal{A}}^{\top}\boldsymbol{\mathcal{A}})^{-1}\boldsymbol{R}^{\top}$, $\operatorname{vec}(\cdot)$ is an operator that stacks the columns of a matrix on top of each other. As a result, the $\{\boldsymbol{X}, \boldsymbol{f}\}$ minimization step reduces to solving the following linear equation system:

$$\boldsymbol{Q} \operatorname{vec}(\boldsymbol{\Lambda}^*) = \boldsymbol{p}, \tag{35}$$

with $LN$ unknown variables. Once $\boldsymbol{\Lambda}^*$ is determined, it is substituted in (32) in order to evaluate the updated abundances. Note that the nonlinear function does not need to be evaluated at each iteration. It can be evaluated once, after the ADMM algorithm has converged.

*2) $\boldsymbol{Z}$* **minimization step:** This step consists of minimizing the augmented Lagrangian with respect to $\boldsymbol{Z}$. After discarding the terms independent of $\boldsymbol{Z}$ in (27) and accounting for the special structure of the matrices $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, and $\boldsymbol{\mathcal{C}}$ given in (25), problem (36) reduces to the following optimization problem:

$$\min_{\boldsymbol{Z}} \quad \tfrac{\rho}{2} \, \|\boldsymbol{Z} - \boldsymbol{X}\|^2_{\mathrm{F}} - \operatorname{tr}(\boldsymbol{\Lambda}_{\rho}^{\top} \boldsymbol{Z}) + \mu \mathcal{J}(\boldsymbol{Z}) + \mathcal{I}_{\mathbb{R}_+^{M \times N}}(\boldsymbol{Z}). \tag{36}$$

In particular, when $\mathcal{J}(\boldsymbol{Z})$ is the Frobenius norm (23), problem (36) reduces to the following positively constrained least squares problem:

$$\min_{\boldsymbol{Z}} \quad \tfrac{1}{2} \, \|\boldsymbol{Z} - \tfrac{\rho}{\rho + \mu} \, (\boldsymbol{X} + \tfrac{1}{\rho} \boldsymbol{\Lambda}_{\rho})\|^2_{\mathrm{F}} + \mathcal{I}_{\mathbb{R}_+^{M \times N}}(\boldsymbol{Z}). \tag{37}$$

The solution of problem (37) is obtained by a projection onto the positive orthant:

$$\boldsymbol{Z} \;\; = \tfrac{\rho}{\rho + \mu} \, (\boldsymbol{X} + \tfrac{1}{\rho} \boldsymbol{\Lambda}_{\rho})_+ \,, \tag{38}$$

where $(\cdot)_+ = \max(0, \cdot)$ is applied component wise. As mentioned previously, $\mathcal{J}(\boldsymbol{Z})$ can be set to a regularization other than the Frobenius norm. For example, we demonstrate the case of the $\ell_1$ norm known for promoting sparse abundances. In this case, problem (36) reduces to the following optimization problem:

$$\min_{\boldsymbol{Z}} \quad \tfrac{1}{2} \, \|\boldsymbol{Z} - (\boldsymbol{X} + \tfrac{1}{\rho} \boldsymbol{\Lambda}_{\rho})\|^2_{\mathrm{F}} + \tfrac{\mu}{\rho} \|\boldsymbol{Z}\|_1 + \mathcal{I}_{\mathbb{R}_+^{M \times N}}(\boldsymbol{Z}). \tag{39}$$

The solution of (39) is the well-known soft thresholding [42] applied to the projection onto the positive orthant:

$$\boldsymbol{Z} = \operatorname{soft}_{\frac{\mu}{\rho}}((\boldsymbol{X} + \frac{1}{\rho} \boldsymbol{\Lambda}_{\rho})_+), \tag{40}$$

where $\operatorname{soft}_{\frac{\mu}{\rho}}(\cdot) = \operatorname{sgn}(\cdot)(|\cdot| - \frac{\mu}{\rho})_+$, and the soft thresholding operator is applied component wise. The solution in equation (40) can be simplified to:

$$\boldsymbol{Z} = (\boldsymbol{X} + \frac{1}{\rho} \boldsymbol{\Lambda}_{\rho} - \frac{\mu}{\rho})_+ \,, \tag{41}$$

where similarly to (38), $(\cdot)_+ = \max(0, \cdot)$ is applied component wise. It is important to note that the sum to one constraint should be relaxed in (24) when the $\ell_1$ norm is considered. Otherwise, $\mathcal{J}(\boldsymbol{Z})$ would be a constant in the feasible set, i.e. when the abundances are positive and sum to one [43].

*3)* **Update of the Lagrange multipliers:** The last step consists of updating the Lagrange multipliers according to the following rule,

$$\boldsymbol{\Lambda}_{\rho}^{\rightarrow} = \boldsymbol{\Lambda}_{\rho} + \rho(\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}), \tag{42}$$

where $\boldsymbol{\Lambda}_{\rho}^{\rightarrow}$ denotes the updated matrix of Lagrange multipliers. This step can be seen as a gradient ascent of the augmented Lagrangian with respect to the Lagrange multiplier. Furthermore, it evaluates the running sum of the constraint residuals.

### C. Implementation details

The ADMM steps described in the previous section are repeated until convergence. As suggested in [39], a reasonable stopping criteria is that the primal and dual residuals must be smaller than some tolerance thresholds, namely,

$$\|\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}\|_{\mathrm{F}} \leq \epsilon_{\text{pri}}, \tag{43}$$

$$\|\rho \boldsymbol{\mathcal{A}}^{\top} \boldsymbol{\mathcal{B}} (\boldsymbol{Z}^{\rightarrow} - \boldsymbol{Z})\|_{\mathrm{F}} \leq \epsilon_{\text{dual}}, \tag{44}$$

The pseudocode of the proposed algorithm is given in Algorithm 1. The most computationally expensive step in the iterative algorithm is the $\boldsymbol{X}$ minimization step which requires solving an $LN \times LN$ system of linear equations (35). Solving the system of linear equations would have a memory complexity $\mathcal{O}(L^2N^2)$ and a runtime complexity $\mathcal{O}(L^3N^3)$. As is, the proposed algorithm is more tractable with multispectral data than with hyperspectral data where the number of bands $L$ is higher. Nevertheless, we do not need to compute the exact solution. The ADMM algorithm will converge even if the $\boldsymbol{X}$ minimization step is carried out only approximately [39], [41]. This allows us to solve (35) using an iterative algorithm which can reduce the runtime complexity and can make the proposed algorithm less computationally expensive even with relatively high values of $L$. The Conjugate Gradient method (CG) [44] is one of the most widely used iterative techniques for solving a large linear system of equations where $\boldsymbol{Q}$ is a positive definite matrix as in (35). The CG can yield the exact solution after $LN$ iterations, but in practice a good initialization yields faster convergence [45]. At each iteration of the CG, the dominating operation is a matrix vector multiplication involving $\boldsymbol{Q}$. In general, the number of operations required for multiplying $\boldsymbol{Q}$ by a vector is $\mathcal{O}(L^2N^2)$.

In the case of the separable kernel, both the memory and the runtime complexity can be reduced furthermore by exploiting the fact that $\boldsymbol{Q}$ is the sum of Kronecker products. In fact, $\boldsymbol{Q}$ can be written as follows:

$$\boldsymbol{Q} = \boldsymbol{I}_L \otimes \boldsymbol{I}_N + \frac{1}{\lambda}\boldsymbol{K} \otimes \boldsymbol{\mathcal{E}} + \frac{1}{\rho}\boldsymbol{I}_N \otimes \boldsymbol{D}, \tag{45}$$

where we have replaced $\widetilde{\boldsymbol{K}}$ by its expression from (14). For an efficient implementation of the product between $\boldsymbol{Q}$ and some vector $\text{vec}(\boldsymbol{\mathcal{J}})$ where $\boldsymbol{\mathcal{J}}$ is an $L \times N$ matrix, the following relationships can be used:

$$\begin{cases} (\boldsymbol{K} \otimes \boldsymbol{\mathcal{E}})\text{vec}(\boldsymbol{\mathcal{J}}) = \text{vec}(\boldsymbol{\mathcal{E}}\boldsymbol{\mathcal{J}}\boldsymbol{K}), \\ (\boldsymbol{I}_N \otimes \boldsymbol{D})\text{vec}(\boldsymbol{\mathcal{J}}) = \text{vec}(\boldsymbol{D}\boldsymbol{\mathcal{J}}) \end{cases} \tag{46}$$

Given (45) and (46), the overall product between $\boldsymbol{Q}$ and some vector $\text{vec}(\boldsymbol{\mathcal{J}})$ is given by:

$$\boldsymbol{Q}\text{vec}(\boldsymbol{\mathcal{J}}) = \text{vec}(\boldsymbol{\mathcal{J}}) + \frac{1}{\lambda}\text{vec}(\boldsymbol{\mathcal{E}}\boldsymbol{\mathcal{J}}\boldsymbol{K}) + \frac{1}{\rho}\text{vec}(\boldsymbol{D}\boldsymbol{\mathcal{J}}). \tag{47}$$

Equation (47) is expressed in terms of ordinary matrix products, which means that we do not have to compute any Kronecker products. Compared to the case where the Kronecker product is evaluated, the memory complexity is reduced from $\mathcal{O}(L^2N^2)$ to $\mathcal{O}(L^2{+}N^2)$ and the number of operations at each iteration is reduced from $\mathcal{O}(L^2N^2)$ to $\mathcal{O}(\max(L^2N, LN^2))$.

Similarly to the previous case, the memory and the runtime complexity of the CG method can be reduced furthermore when using the transformable kernel. Even though $\widetilde{\boldsymbol{K}}$ can not be replaced by a Kronecker product of smaller matrices as in (45), it can be approximated by a product of smaller matrices. For example, this can be done using the Nystrom approximation, see [46] (chap. 18.1) for more details. Nevertheless, it is important to note that the Nystrom approximation introduces errors in the estimation of the Gram matrix. This is in contrast with the first approach, proposed with the separable kernel, where exploiting the properties of the Kronecker product does not alter the exact estimation of the Gram matrix. In order to have a fair comparison between the separable and the transformable kernels, the Nystrom approximation is not used in the experiments.

**Algorithm 1** : $[\boldsymbol{X}, \boldsymbol{F}] = \text{NDU}(\boldsymbol{S}, \boldsymbol{R}, \lambda, \mu, \rho)$

```
Precompute $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{C}}, \widetilde{\boldsymbol{K}}, \boldsymbol{Q}$
Initialize $\boldsymbol{Z}, \boldsymbol{\Lambda}_\rho$
while $\text{res}_{\text{pri}} > \epsilon_{\text{pri}}$ or $\text{res}_{\text{dual}} > \epsilon_{\text{dual}}$ do
    $\boldsymbol{p} = \text{vec}(\boldsymbol{S} + \frac{1}{\rho}\boldsymbol{R}(\boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{A}})^{-1}\boldsymbol{\mathcal{A}}^\top(\boldsymbol{\Lambda}_\rho + \rho(\boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}})))$
    $\text{vec}(\boldsymbol{\Lambda}) = \boldsymbol{Q}^{-1}\boldsymbol{p}$   % See section III-C
    $\boldsymbol{X} = \frac{1}{\rho}(\boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{A}})^{-1}(\boldsymbol{R}^\top\boldsymbol{\Lambda} - \boldsymbol{\mathcal{A}}^\top\boldsymbol{\Lambda}_\rho - \rho\boldsymbol{\mathcal{A}}^\top(\boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}))$
    $\boldsymbol{Z}^{\text{old}} = \boldsymbol{Z}$
    if $\mathcal{J}(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_{\text{F}}^2$ then
        $\boldsymbol{Z} = \frac{\rho}{\rho+\mu}\,(\boldsymbol{X} + \frac{1}{\rho}\boldsymbol{\Lambda}_\rho)_+$
    else if $\mathcal{J}(\boldsymbol{Z}) = \|\boldsymbol{Z}\|_1$ then
        $\boldsymbol{Z} = (\boldsymbol{X} + \frac{1}{\rho}\boldsymbol{\Lambda}_\rho - \frac{\mu}{\rho})_+$
    end if
    $\boldsymbol{\Lambda}_\rho = \boldsymbol{\Lambda}_\rho + \rho(\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}})$
    $\text{res}_{\text{pri}} = \|\boldsymbol{\mathcal{A}}\boldsymbol{X} + \boldsymbol{\mathcal{B}}\boldsymbol{Z} - \boldsymbol{\mathcal{C}}\|_{\text{F}}$
    $\text{res}_{\text{dual}} = \|\rho\boldsymbol{\mathcal{A}}^\top\boldsymbol{\mathcal{B}}(\boldsymbol{Z} - \boldsymbol{Z}^{\text{old}})\|_{\text{F}}$
end while
$\boldsymbol{F} = \frac{1}{\lambda}\widetilde{\boldsymbol{K}}\text{vec}(\boldsymbol{\Lambda})$
$\boldsymbol{F} = \text{reshape}(\boldsymbol{F}, L, N)$
```

## IV. Experiments

### A. Synthetic data: Illustrative examples

*1) Data generation:* The proposed approach is first illustrated using synthetic data. Several patches were generated according to three mixing models that incorporate the main assumptions underlying the proposed nonlinear mixing model, i.e., bilinear contributions, adjacency effects, and band selectivity. The bilinear contributions are created by adding pairwise products of spectra, the adjacency effect is created by adding bilinear contributions from neighboring pixels, and band selectivity is created by assigning a different weight to the nonlinear contributions at different bands. The three mixing models (MM) are denoted as MM 1, MM 2, and MM 3, and they are defined as follows:

- MM 1 (bilinear contributions):

$$\boldsymbol{s}_n \;\; = \boldsymbol{s}_n^{\text{lin}} + u\,\boldsymbol{s}_n^{\text{lin}} \odot \boldsymbol{s}_n^{\text{lin}} + \boldsymbol{e}_n \tag{48}$$

- MM 2 (bilinear contributions + adjacency effects):

$$\boldsymbol{s}_n \;\; = \boldsymbol{s}_n^{\text{lin}} + u \textstyle\sum_{i=n-2}^{n+2} \gamma_{n,i}\,\boldsymbol{s}_i^{\text{lin}} \odot \boldsymbol{s}_i^{\text{lin}} + \boldsymbol{e}_n \tag{49}$$

- MM 3 (bilinear contributions + adjacency effects + band selectivity):

$$\boldsymbol{s}_n \;\; = \boldsymbol{s}_n^{\text{lin}} + u \textstyle\sum_{i=n-2}^{n+2} \gamma_{n,i}\,\boldsymbol{s}_i^{\text{lin}} \odot \boldsymbol{s}_i^{\text{lin}} \odot \boldsymbol{h} + \boldsymbol{e}_n \tag{50}$$

where $u$ in equation (48) is an attenuation parameter set to 0.2 in all the simulations, the coefficients $\gamma_{n,i}$ in equation (49) assign a different weight to the bilinear contributions coming from neighbors, the coeficients were set to $\gamma_{n,n-2} = \gamma_{n,n+2} = 0.05$, $\gamma_{n,n-1} = \gamma_{n,n+1} = 0.3$ and $\gamma_{n,n} = 0.4$, and $\boldsymbol{h}$ in equation (50) is an $L$ dimensional vector where each component assigns a different weight to the nonlinear contribution at the corresponding band. Figure 2 (a) shows the entries of $\boldsymbol{h}$ that were used for the experiments. In fact, $\boldsymbol{h}$ was chosen such that it favors nonlinear contributions at the center of the spectrum and attenuates nonlinear contributions at the extremities of the spectrum. Note that $\boldsymbol{h}$ is unknown by all the unmixing methods used in the experiments. Several patches were created with $N = 100$ pixels using different numbers of endmembers and different values of the SNR. The endmembers were selected from the USGS spectral library of minerals. Their frequency bands are in the range $400 - 2560$ nm, and were decimated such as to have $L = 20$ bands. Figure 2 (b) shows five endmembers spectra used in the simulations. The abundances were generated using a Dirichlet distribution with a unit shape parameter.

*2) Unmixing methods:* We tested three nonlinear unmixing algorithms. The first algorithm is the extended endmember matrix method. It considers the linear mixing model where the endmember matrix is extended by adding the pairwise products of the endmembers [47]. This algorithm is denoted as Ext in the experiments, it consists of solving a positively constrained least squares problem and has no tuning parameters. The second algorithm is based on the scalar RKHS model described in equation (3) and proposed in [21], it is denoted as khype in the experiments. Compared to the proposed optimization problem (22), the abundances and nonlinear function in khype are estimated for every pixel by solving the following optimization problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{a}_n, \Psi_n \in \mathcal{H}_k} & \frac{1}{2}\|\boldsymbol{s}_n - \boldsymbol{R}\boldsymbol{a}_n - \boldsymbol{\Psi}_n(\boldsymbol{R})\|^2 + \frac{\lambda}{2}\|\Psi_n\|^2_{\mathcal{H}_k} \\ & + \frac{\mu}{2}\|\boldsymbol{a}_n\|^2 \\ \text{subject to} & a_{i,n} \succeq 0 \quad \forall\, i = 1, \\ & \sum_{i=1}^{M} a_{i,n} = 1, \end{array} \tag{51}$$

where $\boldsymbol{\Psi}_n(\boldsymbol{R}) = [\Psi_n(\boldsymbol{r}_{\lambda_1}), \ldots, \Psi_n(\boldsymbol{r}_{\lambda_L})]^\top$ and $\Psi_n \in \mathcal{H}_k$ is a scalar valued function in the RKHS $\mathcal{H}_k$ associated with a scalar kernel. Note that the original optimization problem proposed for khype in [21] uses the same parameter to penalize the norm of the abundances and the nonlinear function, i.e. $\lambda = \mu$. Given that NDU requires tuning two parameters $\lambda$ and $\mu$, and in order to have a fair comparison between the two algorithms, the modified khype's optimization problem in (51) has two distinct parameters $\lambda$ and $\mu$. Khype was tested with a Gaussian (G) and a second order homogeneous polynomial (P) kernel. The third algorithm is the one proposed in this paper, it is denoted in what follows as NDU (Nonlinear neighbor and band Dependent Unmixing). Similarly to khype, NDU was tested with a Gaussian (G) and a second order homogeneous polynomial (P) kernel. Furthermore, given that the kernel in NDU is matrix valued, it was tested using a transformable (Tr.) and a separable (Sp.) structure. In the case of the separable kernel, the graph that represents the similarities between the different bands is linear, i.e. each band is connected to the previous and next band, with unit weights. In order to determine $\boldsymbol{v}_n$, the neighborhood was set to $\boldsymbol{\mathcal{C}}_n = \{n, n-1, n+1\}$.

For NDU and khype, the tuning parameters were tested in the range $[10^{-4}\ 10^{-3}\ 10^{-2}\ 10^{-1}\ 1\ 10]$. The standard deviations of the Gaussian kernels used with khype and NDU were chosen such that the resulting Gram matrices have their values in the same range. In particular, we compute all the possible distances between the inputs of the kernel, and set the value of $\sigma$ to the maximum distance found. The polynomial Gram matrices were scaled in order to have their values in the range $[0\ 1]$. Figure 3 shows the Gram matrices used with khype, and NDU in different settings and obtained with $M = 3$ and $SNR = 40$ dB. The first column in Figure 3 shows the $L \times L$ Gram matrices used by khype. The second and third columns in Figure 3 show the Gram matrices used by NDU obtained with a transformable and separable structure respectively. Recall that the NDU Gram matrices are $N \times N$ block matrices, where each block is an $L \times L$ matrix. In the case of the transformable kernel, each block is an $L \times L$ Gram matrix itself. Figure 3 shows that a block or sub-Gram matrix in the transformable kernel is similar the corresponding khype Gram matrix even though it is calculated using the observations themselves rather than the endmember matrix as in khype, whereas for the separable kernel, each block is equal to $\boldsymbol{\mathcal{E}}^\dagger$ multiplied by the corresponding scalar valued kernel.

*3) Performance measures:* The abundance estimation accuracy was evaluated using the root mean square error (RMSE) defined as:

$$\text{RMSE}_{\boldsymbol{X}} = \sqrt{\frac{1}{MN}\|\boldsymbol{X} - \boldsymbol{X}^\star\|^2_{\text{F}}}, \tag{52}$$

where $\boldsymbol{X}$ represents the true abundances matrix and $\boldsymbol{X}^\star$ respresents the abundances estimated by an unmixing algorithm. In addition to the abundances, each one of the unmixing algorithms estimates the nonlinear contributions. Let $\boldsymbol{F}$ and $\boldsymbol{F}^\star$ denote the true and estimated $L \times N$ matrices of nonlinear contributions. The estimation accuracy of $\boldsymbol{F}$ was also evaluated using the RMSE defined as:

$$\text{RMSE}_{\boldsymbol{F}} = \sqrt{\frac{1}{LN}\|\boldsymbol{F} - \boldsymbol{F}^\star\|^2_{\text{F}}}. \tag{53}$$

*4) Simulation results:* Tables I, II and III report the results obtained using MM 1, MM 2, and MM 3 respectively. For each case, we report the root mean square errors of the estimated abundances (first term in brackets) and the estimated nonlinear contributions (second term in brackets). For each set of experiments, namely for each column in the tables, the best scores for the RMSE of the estimated abundances and nonlinear contributions are in bold. Khype and NDU require tuning two parameters, namely $\lambda$ and $\mu$, for concision the parameters are not reported in the tables. The estimation accuracy of the three methods became worse when the noise level increased. The same performance was observed for different values of $M$. In general, Khype outperformed Ext, and NDU outperformed both methods. Note that Ext gave results worse than NDU and khype even when MM 1 was used. This is most probably due to the fact that the sum to one constraint was not incorporated in the implementation of Ext, unlike the case of khype and NDU. In the majority of the cases, NDU gave the best RMSE of the abundances and the nonlinear part when used with the separable and polynomial kernel (Sp. + P). The polynomial

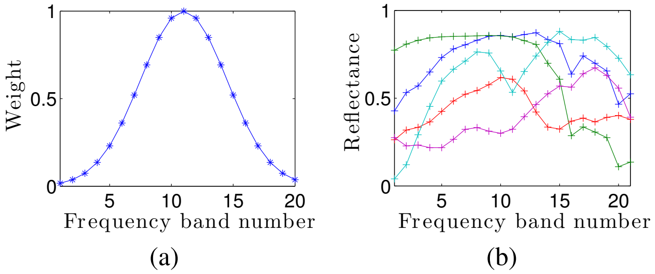


Fig. 2. Fom left to right: (a) The components of vector $\boldsymbol{h}$ corresponding to the weight of the nonlinear contribution at each band, (b) Endmembers spectra used to generate the illustrative examples.

and the Gaussian kernels usually had comparable results in all scenarios. The transformable kernel and the separable kernel had comparable results only when MM 1 and 2 were used, i.e. in tables I and II. However, the separable kernel outperformed the transformable kernel when MM 3 was used, i.e. in table III.

Figure 4 shows the true and estimated nonlinear contributions for all pixels at all bands obtained with $P = 3$ and SNR=40 dB. The first row in Figure 4 shows the true nonlinear contributions. Whereas the following rows show the nonlinear contributions estimated with Ext, khype, NDU with a transformable kernel, and NDU with a separable kernel. For conciseness, we show the results with either the Gaussian or the polynomial kernel for Khype and NDU. In particular, the kernel that gave the best RMSE was chosen. Figure 4 mainly allows to visually compare the estimation of the nonlinear contribution obtained with the various methods and mixing models. The first column in Figure 4 corresponds to the results obtained with MM 1, i.e. with bilinear contributions. It can be seen that khype and NDU slightly outperformed Ext. The second column in Figure 4 corresponds to the results obtained with MM 2, i.e. with bilinear contributions and adjacency effects. The first figure in column 2 shows that the true nonlinear contributions at adjacent pixels (i.e. at adjacent columns in the image) have smooth variations. In this case, NDU with both the transformable and the separable kernel gave the smoothest results compared to Ext and khype. Recall that the kernels used by NDU account for adjacency effects through their input vector $\boldsymbol{v}_n$. The third column in Figure 4 corresponds to the results obtained with MM 3, i.e. with bilinear contributions, adjacency effects, and band selectivity. In accordance with MM 3, the first figure in column 3 shows that the true nonlinear contributions are smooth, they are the most pronounced at the center of the spectrum and they are attenuated (almost zero) at the extremities of the spectrum. In this case, NDU with the separable kernel gave the best results. All the methods estimated the highest nonlinear contributions at the center of the spectrum. However, NDU with the separable kernel handled the attenuation of the bilinear contributions at the extremities of the spectrum better than the other methods. This is probably due to the fact that the prior information on the similarities between the nonlinear contributions at different bands is better in the case of the separable kernel than in the transformable kernel. More precisely, the separable kernel exploits the linear graph structure which promotes smooth variations at adjacent bands (i.e. adjacent rows in the image). Whereas, the transformable kernel exploits the correlations between all bands in order to estimate the nonlinear contributions. Finally, note that the simulations were performed using Matlab on a desktop machine with the following specifications: 2.7 GHz Intel Core i5 processor and 8 GB RAM. With these machine specifications, Ext, khype and NDU took on average $0.41$, $0.11$ and $3.3$ ms per pixel, NDU being the most computationally expensive algorithm.

### B. *Synthetic data: Illustrative example ($L = 200$)*

We repeated the same simulations performed in the previous section with $L = 200$ instead of $L = 20$. Nevertheless, we only used the separable kernel with NDU and used the approach proposed in section III-C to make the algorithm computationally tractable. For concision, we only report the results obtained with $MM = 3$, $M = 3$, and SNR $= 40$ dB. Figure 5 shows the nonlinear part estimated by the three methods, and table IV reports the RMSE of the estimated abundances and nonlinear contributions obtained with each method. In general, the results are similar to the ones obtained with $L = 20$ in the sense that the proposed approach outperformed khype and ExtR and it was able to better account for the smoothness and band-selectivity of the nonlinear contribution. Finally, note that Ext, khype and NDU took on average $1.1$, $2.1$ and $60$ ms per pixel.

### C. *Real data: Gulf of Lion*

*1) Data set description and ground truth:* The second set of experiments considers real data estimated by the Meris spectrometer and captured over the gulf of Lion in the south east of France. The image has $280 \times 330$ pixels, $13$ spectral bands in the range $400 - 800$ nm, and a spatial resolution of $300$ m. This data set will be referred to as the “Meris” image in the following and is depicted in Figure 6 (a). Furthermore, Figure 6 (b) shows the corresponding classification map provided by Corine Land Cover (CLC) database [48]. Note that the two images were coregistered, and the classification map was chosen as close as possible to the date of the Meris image in order to have a consistent comparison. The classification map will be used for visual evaluation, in order to better evaluate and interpret the unmixing results provided by the various algorithms.

The CLC classification map has a spatial resolution approximately $10$ times greater than the Meris data set’s spatial resolution. Therefore, it was downscaled in order to obtain spatial abundance maps for the Meris image [49]. The spatial abundance of a certain class/endmember in a pixel in the estimated low resolution image is the average of the corresponding class occurrences in the corresponding window in the high resolution image. These maps are regarded as a potential visual ground truth that allow to better evaluate and interpret the unmixing results. The first row in Figure 8 shows the ground truth obtained from the CLC classification map, which corresponds to the proportions of three classes: water, agricultural areas, and forests and semi natural areas.

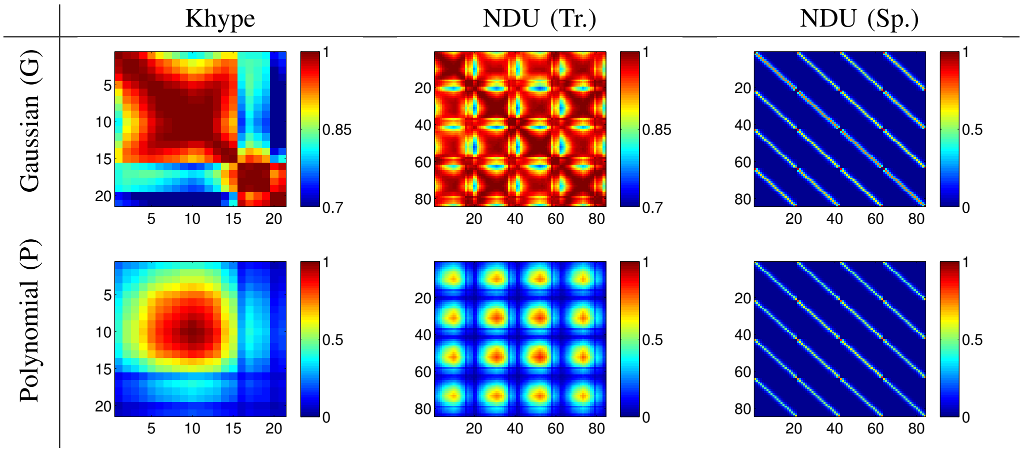


Fig. 3. Gram matrices obtained with $M = 3$ and SNR $= 40$ dB. First column: Gram matrices used by khype, second column: left corners of the Gram matrices obtained using a transformable kernel, and third column: left corners of the Gram matrices obtained using a separable kernel. The first and the second row correspond to the Gaussian and polynomial kernels respectively.

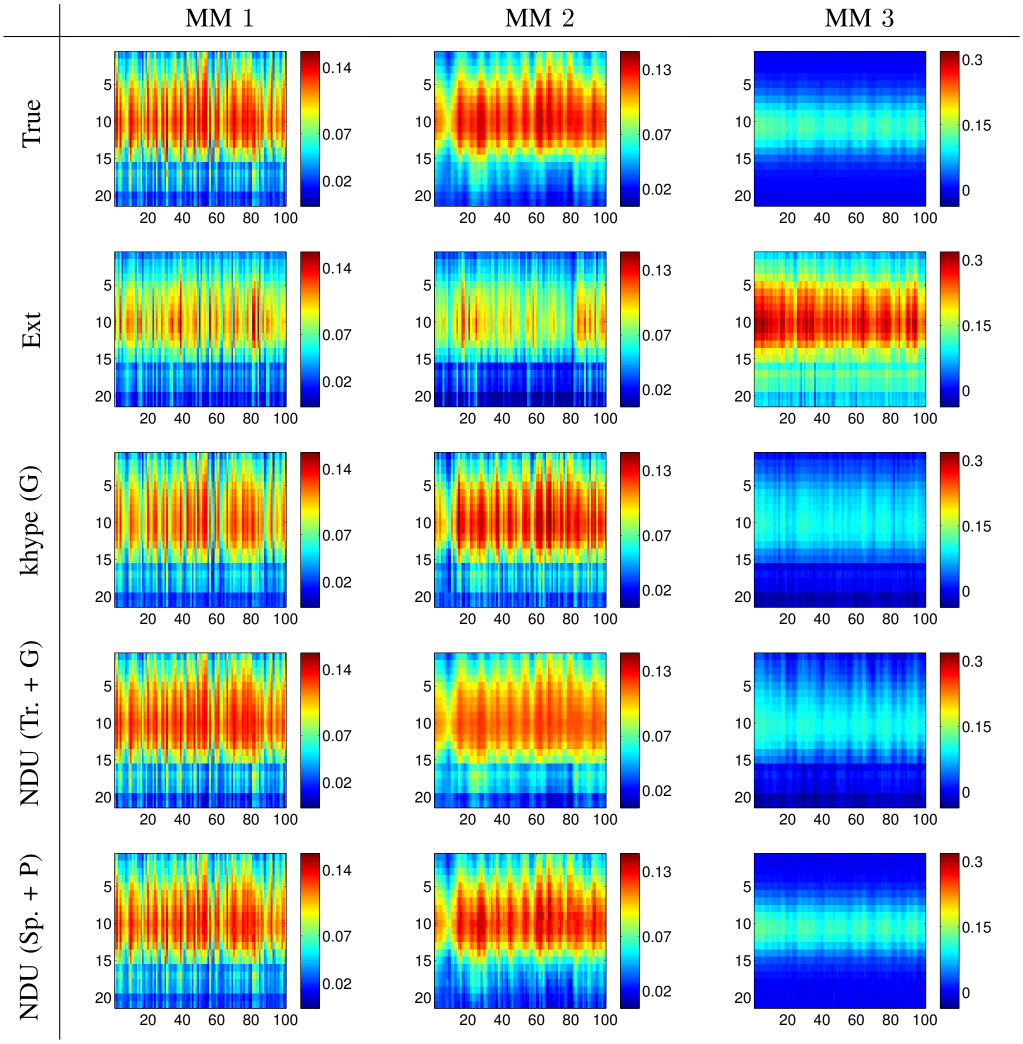


Fig. 4. True and estimated nonlinear contributions in all pixels at all bands obtained with $M = 3$ and SNR=40 dB, the vertical and horizontal axis in each figure represent the frequency band and the pixel number respectively.

TABLE I

RMSE ($\times 10^{-2}$) obtained with MM 1. The two terms in brackets are the RMSE of the estimated abundances (left term) and the nonlinear part (right term).

| | $SNR = 40$ | | | $SNR = 30$ | | | $SNR = 20$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $M = 3$ | $M = 4$ | $M = 5$ | $M = 3$ | $M = 4$ | $M = 5$ | $M = 3$ | $M = 4$ | $M = 5$ |
| Ext | (4.03, 1.69) | (3.91, 2.38) | (3.36, 1.79) | (5.10, 2.86) | (4.48, 2.76) | (4.91, 3.33) | (6.93, 4.99) | (9.50, 7.81) | (9.42, 6.36) |
| Khype (G) | (2.23, 0.74) | (2.40, 1.18) | (2.24, 1.11) | (2.59, 1.33) | (2.80, 1.19) | (2.78, 1.23) | (4.44, 1.89) | (5.25, 1.71) | (4.80, 1.60) |
| Khype (P) | (1.80, 0.631) | (2.06, 0.91) | (1.96, 0.96) | (2.86, 1.08) | (2.60, 1.32) | (2.78, 1.31) | (4.57, 1.79) | (4.69, **1.57**) | (4.34, **1.27**) |
| NDU (Tr. + G) | (1.78, 0.74) | (1.34, 0.55) | (1.21, 0.57) | (1.76, 1.08) | (2.27, **0.97**) | (2.09, **0.78**) | (4.42, 2.28) | (5.34, 2.73) | (4.52, 2.50) |
| NDU (Tr. + P) | (**1.02**, **0.47**) | (**0.81**, **0.25**) | (**1.10**, **0.46**) | (**1.50**, **0.77**) | (**2.23**, 1.02) | (**1.98**, 0.87) | (2.51, 1.63) | (4.82, 2.16) | (5.01, 2.50) |
| NDU (Sp. + G) | (2.65, 1.08) | (3.49, 1.09) | (3.29, 1.79) | (3.40, 2.31) | (4.69, 2.15) | (2.83, 1.72) | (3.33, 6.73) | (4.97, 3.58) | (4.41, 5.95) |
| NDU (Sp. + P) | (2.91, 1.32) | (3.09, 1.52) | (2.87, 1.60) | (3.07, 2.25) | (3.65, 1.94) | (2.57, 1.38) | (3.01, 2.46) | (**4.26**, 6.60) | (**4.27**, 5.49) |

TABLE II

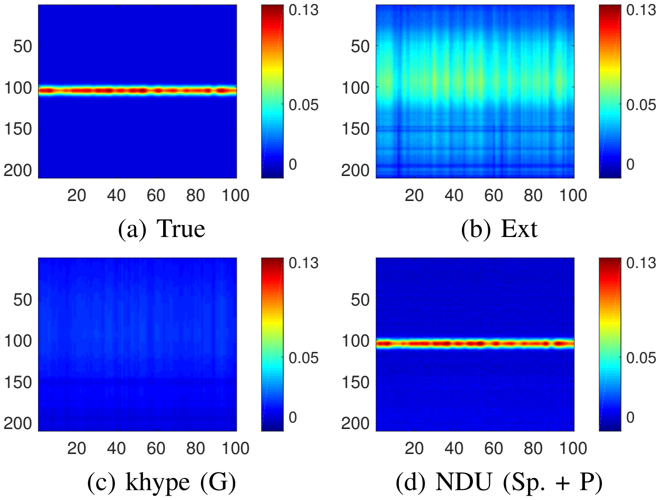
RMSE ($\times 10^{-2}$) obtained with MM 2. The two terms in brackets are the RMSE of the estimated abundances (left term) and the nonlinear part (right term).

| | $SNR = 40$ | | | $SNR = 30$ | | | $SNR = 20$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $M = 3$ | $M = 4$ | $M = 5$ | $M = 3$ | $M = 4$ | $M = 5$ | $M = 3$ | $M = 4$ | $M = 5$ |
| Ext | (2.75, 1.59) | (2.87, 1.42) | (3.12, 1.50) | (3.72, 1.66) | (4.34, 2.88) | (4.05, 3.29) | (6.70, 3.82) | (8.88, 5.95) | (8.20, 6.02) |
| Khype (G) | (1.75, 0.62) | (1.45, 0.63) | (1.20, 0.61) | (2.82, 1.01) | (2.52, 0.82) | (2.29, 1.00) | (4.40, 1.90) | (5.26, 1.57) | (**5.27**, **2.05**) |
| Khype (P) | (1.62, 0.60) | (1.31, 0.55) | (1.12, 0.53) | (2.56, 0.86) | (2.61, 0.89) | (2.26, 1.05) | (4.57, **1.70**) | (4.73, **1.51**) | (5.59, 2.21) |
| NDU (Tr. + G) | (**0.84**, **0.58**) | (**0.90**, 0.36) | (**0.86**, 0.40) | (**1.23**, **0.54**) | (**1.90**, **0.52**) | (1.87, 0.70) | (8.04, 3.82) | (7.25, 3.42) | (6.21, 3.60) |
| NDU (Tr. + P) | (1.43, 0.60) | (1.11, **0.33**) | (0.99, **0.37**) | (2.22, 0.72) | (2.15, 0.43) | (**1.82**, **0.61**) | (4.98, 2.66) | (6.47, 2.89) | (5.60, 2.81) |
| NDU (Sp. + G) | (2.20, 0.91) | (5.08, 1.71) | (3.23, 1.46) | (4.87, 2.51) | (4.11, 1.90) | (3.10, 2.07) | (4.37, 6.94) | (4.11, 3.44) | (6.06, 3.69) |
| NDU (Sp. + P) | (1.61, 0.63) | (2.61, 1.07) | (2.04, 1.18) | (2.75, 1.76) | (2.62, 1.17) | (2.24, 1.53) | (**3.74**, 6.99) | (**4.00**, 2.26) | (5.56, 2.59) |

TABLE III

RMSE ($\times 10^{-2}$) obtained with MM 3. The two terms in brackets are the RMSE of the estimated abundances (left term) and the nonlinear part (right term).

| | $SNR = 40$ | | | $SNR = 30$ | | | $SNR = 20$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | $M = 3$ | $M = 4$ | $M = 5$ | $M = 3$ | $M = 4$ | $M = 5$ | $M = 3$ | $M = 4$ | $M = 5$ |
| Ext | (3.71, 5.65) | (7.45, 5.67) | (5.89, 5.67) | (4.40, 6.17) | (8.59, 6.06) | (7.04, 5.84) | (7.47, 6.66) | (10.1, 8.79) | (10.2, 10.3) |
| Khype (G) | (0.44, 2.47) | (2.67, 1.95) | (2.84, 1.53) | (1.20, 2.57) | (3.21, 2.04) | (2.89, 1.51) | (3.45, 2.58) | (5.85, 1.98) | (5.22, 1.70) |
| Khype (P) | (1.11, 2.43) | (3.33, 1.93) | (3.04, 1.54) | (1.51, 2.52) | (3.63, 2.01) | (3.14, 1.52) | (2.94, 2.46) | (5.87, 1.93) | (5.21, 1.64) |
| NDU (Tr. + G) | (3.63, 2.28) | (5.92, 2.24) | (4.55, 2.02) | (6.09, 3.12) | (7.39, 2.79) | (5.06, 2.27) | (9.63, 4.47) | (10.4, 4.14) | (8.81, 4.22) |
| NDU (Tr. + P) | (6.54, 3.19) | (5.79, 2.46) | (5.02, 2.49) | (7.90, 3.66) | (7.30, 3.02) | (5.29, 2.69) | (10.0, 4.52) | (10.0, 4.06) | (8.98, 4.56) |
| NDU (Sp. + G) | (0.51, 0.43) | (1.05, 0.50) | (0.90, 0.61) | (1.20, 1.49) | (1.61, 1.28) | (**2.41**, 1.06) | (2.54, **3.41**) | (**5.10**, 0.92) | (4.93, 0.92) |
| NDU (Sp. + P) | (**0.41**, **0.29**) | (**0.90**, **0.47**) | (**0.80**, **0.46**) | (**1.07**, **0.93**) | (**1.54**, **0.89**) | (2.47, **1.04**) | (**2.53**, **3.81**) | (**5.10**, **0.90**) | (**4.89**, **0.91**) |



(a) True

(b) Ext

(c) khype (G)

(d) NDU (Sp. + P)

Fig. 5. True and estimated nonlinear contributions in all pixels at all bands obtained with $M = 3$ and SNR=40 dB, the vertical and horizontal axis in each figure represent the frequency band and the pixel number respectively.

TABLE IV

Root mean square error RMSE ($\times 10^{-2}$) of the abundances and nonlinear contribution obtained with the synthetic data using $M = 4$ and $L = 200$ for $SNR = 40$ dB.

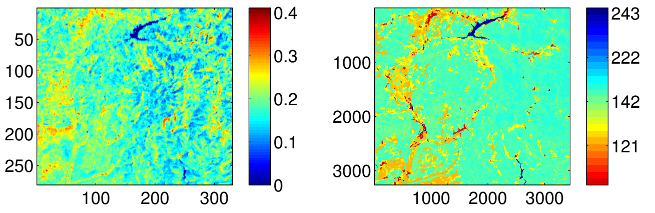
| | RMSE$_{\boldsymbol{X}}$ | RMSE$_{\boldsymbol{F}}$ |
|---|---|---|
| ExtR | 1.97 | 3.1 |
| Khype (G) | 0.86 | 1.86 |
| Khype (P) | 1.37 | 1.89 |
| NDU (Sep. + G) | 0.55 | 0.30 |
| NDU (Sep. + P) | 0.43 | 0.21 |

2) *Unmixing results:* We extracted 3 and 4 endmembers using virtual component analysis VCA [50]. We noticed that each time one of the endmembers extracted by VCA was not meaningful in the sense that the corresponding abundance map was not spatially coherent. Nevertheless, in the case with 4 extracted endmembers the three abundance maps corresponding to the meaningful endmembers were relatively in accordance with the estimated ground truth. In what follows we only show

the results obtained with four endmembers. Figure 7 shows the estimated endmembers spectra, note that endmember 4 corresponds to the outlier. The abundance maps for each endmember were estimated using the fully constrained least squares approach (FCLS), Ext, khype and NDU. The separable kernel was used with NDU with a linear graph as in the experiments with synthetic data. Both khype and NDU were tested using a Gaussian and a second order homogeneous polynomial kernel. As in the previous section, $\boldsymbol{v}_n$ was defined using the pixels and its neighboring pixels spectra. In particular, the four neighbors were chosen, namely the upper, lower, left and right neighbors of each pixel. The tuning parameters $\lambda$ and $\mu$ were set to 10 and $10^{-4}$ respectively for both khype and NDU. Unlike khype and Ext that were applied on each pixel separately, the image was divided into $10 \times 10$ patches and NDU was applied on each patch.

Table V reports the root mean square error (RMSE) and the average spectral angle (ASA) of the reconstructed image using each algorithm. The RMSE is computed for all the observations in the image, and the ASA is given in radian. Table V shows that NDU scored the best results in terms of both the RMSE and the ASA. The results obtained with Ext slightly improved the ones obtained with FCLS. Khype outperformed both methods, Ext and FCLS, and had results very close to the ones obtained with NDU. Figure 8 shows that the abundance maps estimated by the various algorithms are rather similar. As mentioned previously, the fourth endmember corresponds to noise hence its abundance maps are not shown in Figure 8. Figure 9 shows the nonlinear part estimated by NDU and khype at band 10. In fact, most of the areas where nonlinear contributions appear are mainly located on the boundaries of agricultural areas (endmember 2) surrounded by forests and semi-natural areas (endmember 3). Note that the nonlinear contributions estimated by NDU are relatively spatially smoother than the ones estimated by khype. However, NDU results exhibit some artifacts due to the fact that the image was partitioned into square patches.

Finally, Figure 10 compares the nonlinear contribution estimated by NDU and khype at all bands for the pixels delimited by rows 171 and 180 and columns 231 and 240. Both algorithms estimated the highest nonlinear contributions for this particular region. However, the nonlinear contributions have different variations throughout the spectral bands. NDU estimated the highest nonlinear contributions at the higher frequency bands whereas khype estimated almost the same level of nonlinearity at all frequency bands. Furthermore, NDU estimated smooth nonlinear contributions at adjacent pixels compared to khype. It can be concluded that NDU captures more spectral variability throughout the spectral bands and that it provides smooth nonlinear contributions at adjacent pixels.



(a) Band 10 of Meris data

(b) Classification map

Fig. 6. Fom left to right: (a) band 10 of the Meris data set, (b) corresponding CLC classification map (See classes names in Appendix A).

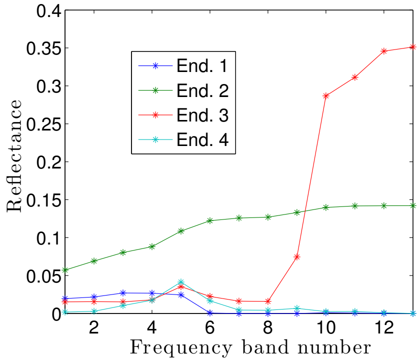


Fig. 7. Endmembers spectra estimated by VCA for the Meris data set.

## V. Conclusion

This paper proposed a new kernel based nonlinear mixing model for hyperspectral data. The proposed vector-valued function is able to account for band dependent and neighboring nonlinear contributions. The proposed framework has several characteristics. It allows to handle in a unified framework several types of nonlinearities depending on the choice of the kernel function. Unlike nonlinear models proposed in the literature, it considers a different nonlinear function for each spectral band. The fact that the nonlinear function acts on the reflectance vectors observed in the corresponding pixel and its neighbors is intended to account for nonlinearities originating from the ground cover of the pixel and its neighbors. Furthermore, the separable kernel design can be used to incorporate prior information regarding the similarities between nonlinear contributions at different bands. In particular, a linear graph was proposed to promote smooth nonlinear variations between adjacent bands. The performance of the proposed approach was validated on synthetic and real data estimated by the Meris spectrometer and captured over the gulf of Lion in the south east of France. Finally, note that the proposed approach requires partitioning the image into patches which can result in artifacts in the estimated nonlinear part. Future work should aim at attenuating those artifacts through an extension of the vector-valued approach. More precisely, the vector-valued framework can be adapted such as to promote smoothness between the estimated nonlinear contributions at adjacent bands and between the nonlinear functions at adjacent patches simultaneously. Furthermore, the proposed approach requires the extraction of the endmembers beforehand using some endmember extraction algorithm. Future work should aim at extending the proposed algorithm to the unsupervised case where the endmembers are jointly estimated with the abundances and nonlinear function.

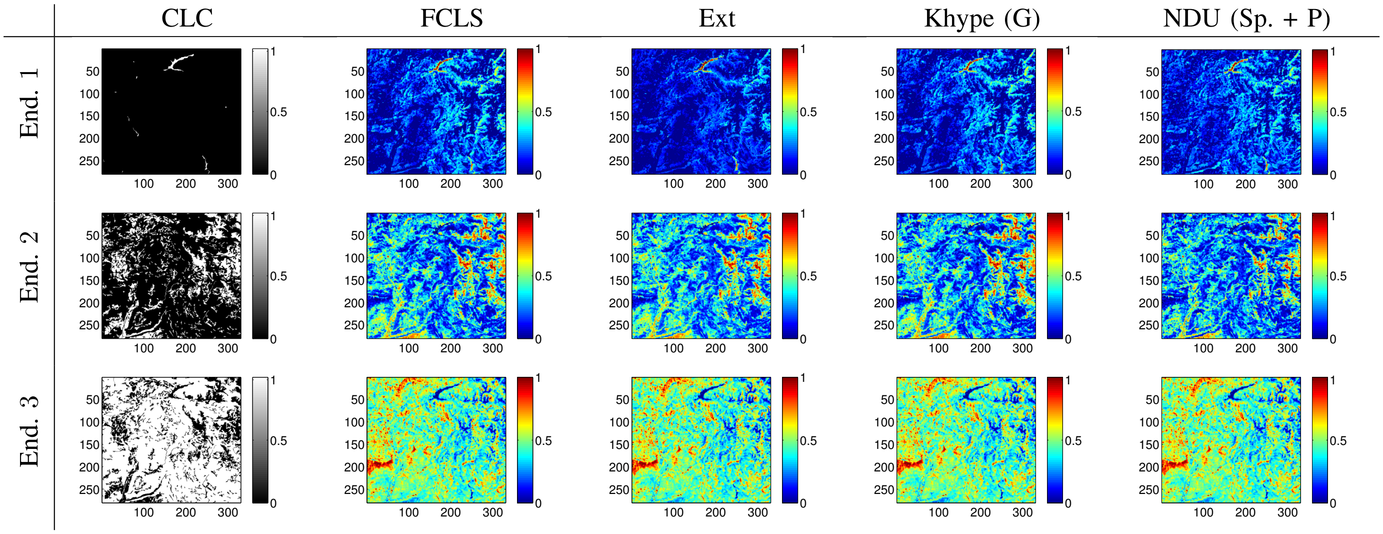
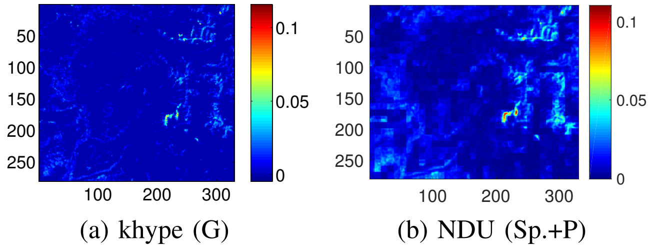
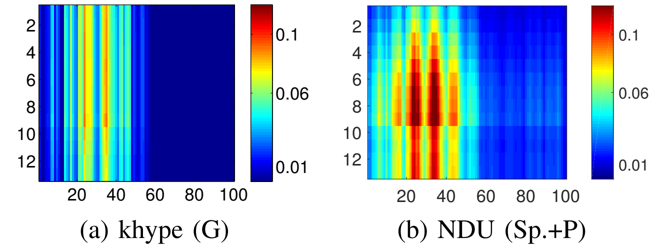


Fig. 8. Abundance maps of the first three endmembers obtained with VCA and corresponding to the Meris real data set. The abundance maps of End. 1, 2, and 3 correspond to water, agricultural areas, and forests and semi natural areas respectively.



(a) khype (G) (b) NDU (Sp.+P)

Fig. 9. Nonlinear contributions at all pixels at band 10 obtained with: (a) khype used with a Gaussian kernel (G) and (b) NDU used with a separable and polynomial kernel (Sp.+P).



(a) khype (G) (b) NDU (Sp.+P)

Fig. 10. Nonlinear contributions at all bands in some of the pixels in the Meris data set. The horizontal and vertical axis correspond to the pixel and frequency index respectively.

TABLE V
ROOT MEAN SQUARE ERROR RMSE ($\times 10^{-2}$) AND AVERAGE SPECTRAL ANGLE (ASA) IN RADIAN OF THE RECONSTRUCTED SPECTRA OBTAINED WITH THE MERIS DATA SET.

| | RMSE | ASA |
|---|---|---|
| FCLS | 1.14 | 0.0393 |
| Ext | 1.13 | 0.0343 |
| Khype (G) | 0.66 | 0.0338 |
| Khype (P) | 1.04 | 0.0381 |
| NDU (Sep. + G) | 0.41 | 0.0152 |
| NDU (Sep. + P) | 0.45 | 0.0204 |

## REFERENCES

[1] G. Shaw and H. Burke, “Spectral imaging for remote sensing,” *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 3–28, 2003.

[2] B. Hapke, *Theory of reflectance and emittance spectroscopy*, Cambridge University Press, 2012.

[3] C. Chang, *Hyperspectral data exploitation: theory and applications*, John Wiley and Sons, 2007.

[4] J. Adams, M. Smith, and P. Johnson, “Spectral mixture modeling: a new analysis of rock and soil types at the viking lander 1 site,” *Journal of Geophysical Research: Solid Earth*, vol. 91, no. B8, pp. 8098–8112, 1986.

[5] N. Keshava and J. F. Mustard, “Spectral unmixing,” *IEEE Transactions on Signal Processing*, vol. 19, no. 1, pp. 44–57, 2002.

[6] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[7] D. C. Heinz and C. I. Chang, “Fully constrained least squares linear spectral mixture analysis method for material quantification in hypersectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 3, pp. 529–545, 2001.

[8] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. C. Bermudez, S. McLaughlin, and A. Hero, “Nonlinear unmixing of hyperspectral images: Models and algorithms,” *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 82–94, 2014.

[9] R. Heylen, M. Parente, and P. Gader, “A review of nonlinear hyperspectral unmixing methods,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 1844 –1868, 2014.

[10] J. Nascimento and J. Bioucas-Dias, “Nonlinear mixture model for hyperspectral unmixing,” in *SPIE*, 2009, pp. 74770I–74770I.

[11] W. Fan, B. Hu, J. Miller, and M. Li, “Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated-forest hyperspectral data,” *International Journal of Remote Sensing*, vol. 30, no. 11, pp. 2951–2962, 2009.

[12] A. Halimi, Y. Altmann, N. Dobigeon, and J-Y Tourneret, “Nonlinear unmixing of hyperspectral images using a generalized bilinear model,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4153–4162, 2011.

[13] Y. Altmann, N. Dobigeon, and J.-Y. Tourneret, “Unsupervised post-nonlinear unmixing of hyperspectral images using a hamiltonian monte carlo algorithm,” *IEEE Transactions on Image Processing*, vol. 23, no. 6, pp. 2663–2675, 2014.

[14] Y. Altmann, A. Halimi, N. Dobigeon, and J-Y Tourneret, “Supervised nonlinear spectral unmixing using a postnonlinear mixing model for hyperspectral imagery,” *IEEE Transaction on Image Processing*, vol. 21, no. 6, pp. 3017–3025, 2012.

TABLE VI
CLASSES IN CLASSIFICATION MAP OF MERIS DATA SET.

| Code | Color | description |
|---|---|---|
| **1** | | **Artificial surfaces** |
| **1.1** | | **Urban fabric** |
| 1.1.1 | 🟥 | Continuous urban fabric |
| 1.1.2 | 🟥 | Discontinuous urban fabric |
| **1.2** | | **Industrial, commercial and transport units** |
| 1.2.1 | 🟥 | Industrial or commercial units |
| 1.2.2 | 🟥 | Road and rail networks and associated land |
| 1.2.4 | 🟧 | Airports |
| **1.3** | | **Mine, dump and construction sites** |
| 1.3.1 | 🟧 | Mineral extraction sites |
| **1.4** | | **Artificial, non-agricultural vegetated areas** |
| 1.4.2 | 🟧 | Sport and leisure facilities Not translated |
| **2** | | **Agricultural areas** |
| **2.1** | | **Arable land** |
| 2.1.1 | 🟧 | Non-irrigated arable land |
| **2.2** | | **Permanent crops** |
| 2.2.1 | 🟨 | Vineyards |
| 2.2.2 | 🟨 | Fruit trees and berry plantations |
| **2.3** | | **Pastures** |
| 2.3.1 | 🟨 | Pastures |
| **2.4** | | **Heterogeneous agricultural areas** |
| 2.4.2 | 🟩 | Complex cultivation patterns |
| 2.4.3 | 🟩 | Land principally occupied by agriculture |
| 2.4.4 | 🟩 | Agro-forestry areas |
| **3** | | **Forest and seminatural areas** |
| **3.1** | | **Forests** |
| 3.1.1 | 🟩 | Broad-leaved forest |
| 3.1.2 | 🟩 | Coniferous forest |
| 3.1.3 | 🟦 | Mixed forest |
| **3.2** | | **and/or herbaceous vegetation associations** |
| 3.2.1 | 🟦 | Natural grasslands |
| 3.2.2 | 🟦 | Moors and heathland |
| 3.2.3 | 🟦 | Sclerophyllous vegetation |
| 3.2.4 | 🟦 | Transitional woodland-shrub |
| **3.3** | | **Open spaces with little or no vegetation** |
| 3.3.1 | 🟦 | Beaches, dunes, sands |
| 3.3.2 | 🟦 | Bare rocks |
| 3.3.3 | 🟦 | Sparsely vegetated areas |
| 3.3.5 | 🟦 | Glaciers and perpetual snow |
| **5** | | **Water bodies** |
| **5.1** | | **Inland waters** |
| 5.1.1 | 🟦 | Water courses |
| 5.1.2 | 🟦 | Water bodies |

[15] N. Dobigeon, L. Tits, B. Somers, Y. Altmann, and P. Coppin, "A comparison of nonlinear mixing models for vegetated areas using simulated and real hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 1869–1878, 2014.

[16] I. Meganem, P. Deliot, X. Briottet, Y. Deville, and S. Hosseini, "Linear–quadratic mixing model for reflectances in urban environments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 1, pp. 544–558, 2014.

[17] R. Heylen and P. Scheunders, "A multilinear mixing model for nonlinear spectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, 2016.

[18] N. Aronszajn, "Theory of reproducing kernels," *Transactions of the American mathematical society*, pp. 337–404, 1950.

[19] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*, Cambridge university press, 2004.

[20] V. Vapnik, *The nature of statistical learning theory*, Springer Science, 2013.

[21] J. Chen, C. Richard, and P. Honeine, "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 480–492, 2013.

[22] N. Yokoya, J. Chanussot, and A. Iwasaki, "Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 2, pp. 1430–1437, 2014.

[23] C. Févotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4810–4819, 2015.

[24] B. Somers, K. Cools, S. Delalieux, J. Stuckens, D. Van der Zande, W. Verstraeten, and P. Coppin, "Nonlinear hyperspectral mixture analysis for tree cover estimates in orchards," *Remote Sensing of Environment*, vol. 113, no. 6, pp. 1183–1193, 2009.

[25] R. Richter, M. Bachmann, W. Dorigo, and A. Muller, "Influence of the adjacency effect on ground reflectance measurements," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 565–569, 2006.

[26] D. Tanre, P. Deschamps, P. Duhaut, and M. Herman, "Adjacency effect produced by the atmospheric scattering in thematic mapper data," *Journal of Geophysical Research: Atmospheres*, vol. 92, no. D10, pp. 12000–12006, 1987.

[27] D. Burazerovic, R. Heylen, B. Geens, S. Sterckx, and P. Scheunders, "Detecting the adjacency effect in hyperspectral imagery with spectral unmixing techniques," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 3, pp. 1070–1078, 2013.

[28] S. Liang, H. Fang, and M. Chen, "Atmospheric correction of landsat etm+ land surface imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 11, pp. 2490–2498, 2001.

[29] D. Hadjimitsis, G. Papadavid, A. Agapiou, K. Themistocleous, M. Hadjimitsis, A. Retalis, S. Michaelides, N. Chrysoulakis, L. Toulios, and C. Clayton, "Atmospheric correction for satellite remotely sensed data intended for agricultural applications: impact on vegetation indices," *Natural Hazards and Earth System Science*, vol. 10, no. 1, pp. 89–95, 2010.

[30] C. Micchelli and M. Pontil, "On learning vector-valued functions," *Neural Computation*, vol. 17, no. 1, pp. 177–204, 2005.

[31] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in computational mathematics*, vol. 13, no. 1, pp. 1–50, 2000.

[32] M. Alvarez, L. Rosasco, and N. Lawrence, "Kernels for vector-valued functions: A review," *Foundations and Trends in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[33] T. Evgeniou, C. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," in *Journal of Machine Learning Research*, 2005, pp. 615–637.

[34] L. Baldassarre, *Multi-output learning with spectral filters*, Ph.D. thesis, Ph. D. dissertation, Dept. of Physics-University of Genoa, 2010.

[35] L. J. Grady and J. R. Polimeni, *Discrete calculus*, Springer, 2010.

[36] G. Asner and D. Lobell, "A biogeophysical approach for automated swir unmixing of soils and vegetation," *Remote sensing of environment*, vol. 74, no. 1, pp. 99–112, 2000.

[37] J. Chen, C. Richard, and P. Honeine, "Nonlinear estimation of material abundances in hyperspectral images with l1-norm spatial regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2654 – 2665, 2014.

[38] R. Heylen, D. Burazerovic, and P. Scheunders, "Fully constrained least squares spectral unmixing by simplex projection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4112–4122, 2011.

[39] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[40] J. Esser, *Primal dual algorithms for convex models and applications to image restoration, registration and nonlocal inpainting*, Ph.D. thesis, University of California Los Angeles, 2010.

[41] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[42] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[43] J. Bioucas-Dias and M. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*. IEEE, 2010, pp. 1–4.

[44] Y. Saad, *Iterative methods for sparse linear systems*, Siam, 2003.

[45] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," 1994.

[46] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Press Syndicate of the University of Cambridge, second edition, 2002.

[47] N. Raksuntorn and Q. Du, "Nonlinear spectral mixture analysis for hyperspectral imagery in an unknown environment," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 836–840, 2010.

[48] Ministry of ecology sustainable development and energy, “www.statistiques.developpement-durable.gouv.fr/clc/carte/metropole,” .

[49] B. Zhukov, D. Oertel, F. Lanzl, and G. Reinhackel, “Unmixing-based multisensor multiresolution image fusion,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, pp. 1212–1226, 1999.

[50] J. M. P. Nascimento and J. M. Bioucas-Dias, “Vertex component analysis: A fast algorithm to unmix hyperspectral data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.

**Rita Ammanouil** (S’13) was born in Beirut, Lebanon in January 1991. She received the bachelor in Computer and Communication Engineering from the Holly Spirit University of Kaslik, Lebanon, in 2013, and the Ph.D. degree from the University of Nice Sophia Antipolis, in 2016. She is currently a Post-Doctoral researcher within the Joseph-Louis Lagrange Laboratory (CNRS, OCA). Her research interests are Hyperspectral image unmixing and reconstruction.

**André Ferrari** (SM’91-M’93) received the engineering degree from École Centrale de Lyon, France, in 1988 and the M.Sc. and Ph.D. degrees from the University of Nice Sophia Antipolis (UNS), France, in 1989 and 1992, respectively, all in electrical and computer engineering. He is currently a Professor at UNS. He is a member of the Joseph-Louis Lagrange Laboratory (CNRS, OCA), where his research activity is centered around statistical signal processing and modeling, with a particular interest in applications to astrophysics.

**Cédric Richard** (S’98–M’01–SM’07) received the Dipl.-Ing. and the M.S. degrees in 1994, and the Ph.D. degree in 1998, from Compiègne University of Technology, France, all in electrical and computer engineering. He is a Full Professor at the Université Côte d’Azur, France. He was a junior member of the Institut Universitaire de France in 2010-2015. His current research interests include statistical signal processing and machine learning. Cédric Richard is the author of over 250 papers. He was the General Co-Chair of the IEEE SSP Workshop that was held in Nice, France, in 2011. He was the Technical Co-Chair of EUSIPCO 2015 that was held in Nice, France, and of the IEEE CAMSAP Workshop 2015 that was held in Cancun, Mexico. He serves as a Senior Area Editor of the IEEE Transactions on Signal Processing and as an Associate Editor of the IEEE Transactions on Signal and Information Processing over Networks since 2015. He is also an Associate Editor of Signal Processing Elsevier since 2009. Cédric Richard is member of the Machine Learning for Signal Processing (MLSP TC) Technical Committee, and served as member of the Signal Processing Theory and Methods (SPTM TC) Technical Committee in 2009-2014.

**Sandrine Mathieu** received the Ph.D. degree. She is an Expert Engineer at the Thales Alenia Space Cannes, Cannes, France, in spatial system dimensioning from user requirements. She has been working on Sentinel 3 and MTG Programs as responsible of image quality technical specifications. She is used to spatial data fusion. She is currently the Coordinator of the French Government initiative Booster PACA, which promotes innovative applications from Copernicus Data and Services.