






Emulation as an Accurate Alternative to Interpolation in Sampling Radiative Transfer Codes

Jorge Vicent , Jochem Verrelst , Juan Pablo Rivera-Caicedo, Neus Sabater, Jordi Muñoz-Marí , *Member, IEEE*, Gustau Camps-Valls , *Fellow, IEEE*, and José Moreno , *Member, IEEE*

Abstract—Computationally expensive radiative transfer models (RTMs) are widely used to realistically reproduce the light interaction with the earth surface and atmosphere. Because these models take long processing time, the common practice is to first generate a sparse look-up table (LUT) and then make use of interpolation methods to sample the multidimensional LUT input variable space. However, the question arises whether common interpolation methods perform most accurately. As an alternative to interpolation, this paper proposes to use emulation, i.e., approximating the RTM output by means of the statistical learning. Two experiments were conducted to assess the accuracy in delivering spectral outputs using interpolation and emulation: at canopy level, using PROSAIL; and at top-of-atmosphere level, using MODTRAN. Various interpolation (nearest-neighbor, inverse distance weighting, and piece-wise linear) and emulation [Gaussian process regression (GPR), kernel ridge regression, and neural networks] methods were evaluated against a dense reference LUT. In all experiments, the emulation methods clearly produced more accurate output spectra than classical interpolation methods. The GPR emulation performed up to ten times more accurately than the best performing interpolation method, and this with a speed that is competitive with the faster interpolation methods. It is concluded that emulation can function as a fast and more accurate alternative to commonly used interpolation methods for reconstructing RTM spectral data.

Index Terms—Emulation, interpolation, look-up tables (LUT), machine learning, performance simulators, processing speed, radiative transfer models (RTMs).

Manuscript received January 27, 2018; revised May 23, 2018 and August 21, 2018; accepted October 4, 2018. This work was carried out in the frame of ESA's project FLEX L2 End-to-End Simulator Development and Mission Performance Assessment under ESA Contract 4000119707/17/NL/MP. The work of J. Verrelst was supported by the European Research Council under the ERC-2017-STG SENTIFLEX project Grant 755617. The work of N. Sabater was supported by the MINECO under Grant ESP2016-79503-C2-1-P for the project "AvanFLEX: Advanced products for the FLEX mission. The work of J. Muñoz-Marí was supported by the MINECO/ERDF under Grant CICYT TIN2015-64210-R. The work of G. Camps-Valls was supported by the ERC under the ERC-CoG-2014 SEDAL project Grant 647423. (*Corresponding author: Jorge Vicent.*)

J. Vicent, J. Verrelst, N. Sabater, J. Muñoz-Marí, G. Camps-Valls, and J. Moreno are with the Image Processing Laboratory, University of Valencia, Valencia 46980, Spain (e-mail: jorge.vicent@uv.es; jochem.verrelst@uv.es; m.neus.sabater@uv.es; jordi.munoz@uv.es; gustau.camps@uv.es; jose.moreno@uv.es).

J. P. Rivera-Caicedo is with the CONACYT-UAN, Departamento: Secretaría de investigación y posgrado, 63155, Tepic, Mexico (e-mail: jprivera@conacyt.mx).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JSTARS.2018.2875330

I. INTRODUCTION

PHYSICALLY-BASED radiative transfer models (RTMs) allow remote sensing scientists to understand the light interactions between water, vegetation, and atmosphere [1]–[3]. RTMs are physically-based computer models that describe scattering, absorption, and emission processes in the visible to microwave region [4], [5]. These models are widely used in applications, such as inversion of atmospheric and vegetation properties from remotely sensed data (see [6] for a review), to generate artificial scenes as would be observed by a sensor [7]–[9], and sensitivity analysis of RTMs [10]. In the optical domain, a diversity of vegetation, atmosphere, and water RTMs have continuously been improved in accuracy from simple semiempirical RTMs toward advanced ray tracing RTMs. This evolution has led to an increase in complexity, interpretability, and computational requirements to run the model, which bears implications toward practical applications. On the one hand, computationally cheap RTMs are models with relatively few input parameters that enables fast calculations (e.g., [11] and [12] for vegetation and [13], [14] for atmosphere). On the other hand, computationally expensive RTMs are complex physically-based mathematical models with a large number of input variables. In short, the following families of RTMs can be considered as computationally expensive: Monte Carlo ray tracing models (e.g., Raytran [15], FLIGHT [16], and librat [17]), voxel-based models (e.g., DART [18]), and advanced integrated vegetation and atmospheric transfer models that consists of various sub-routines (e.g., SimSphere [19], SCOPE [20], and MODTRAN [21]). Despite the higher accuracy of these RTMs to model the light-vegetation and atmosphere interactions (see e.g., [15] and [22]), their high computational burden make them impractical for practical applications that demand many simulations, and alternatives have to be sought.

In order to overcome this limitation, RTMs are most commonly applied by means of look-up tables (LUTs) [6]. LUTs are pre-stored RTM output data so that the computational of the RTM has to be done only one time, prior to the application. Nevertheless, for reasons of memory storage and processing time, LUTs are usually kept to a reasonable size, especially in case of computationally expensive RTMs. The common approach is then to seek through the multidimensional LUT input variable space by means of interpolation techniques. Various interpolation techniques have been developed both for gridded and scattered datasets [23]–[25]. Linear interpolation is the most used approach in both gridded and scattered datasets due

to its balance between processing speed and accuracy [26]–[28]. However, the main drawback of the linear interpolation in high-dimensional scattered datasets is that the underlying triangulation is computationally expensive and uses large computer memory.

Emulation of costly codes is an alternative to interpolation, but based on statistical principles. The core idea of an emulator is to extract (or *learn*) the statistical information from a limited set of simulations of the original deterministic model [29], [30]. Emulators then approximate the original RTM at a tiny fraction of its speed and this can be readily applied in tedious processing routines [31], [32]. The use of emulators deals with some extra advantages such as the use of a nongridded input parameter space, making it more versatile than several interpolation methods (e.g., piece-wise cubic splines). In different research fields, such as engineering, energy, robotics, and environmental modeling, emulators have already demonstrated to be a more efficient alternative to classical LUT interpolation methods [29], [33]–[38]. However, these studies are generally limited to models with a few output dimensions, low levels of noise, or low degrees of freedom and ill-posedness. Therefore, an important question arises here whether emulators are able to compete with interpolation methods in sampling capabilities of hyperspectral RTM outputs, both in terms of accuracy and processing speed. The problem is thus new and actually challenges the potential capabilities of emulators. This brings us to the main objective of this paper, i.e., to analyze the performance of emulators as an alternative of classical RTM-based LUT interpolation for sampling the LUT parameter space. To do so, two contrasting LUT spectral outputs are examined: one of spectrally smooth top-of-canopy (TOC) reflectance data, and another of more sharper (high frequency bands) top-of-atmosphere (TOA) radiance data. We give experimental evidence that emulation generally outperforms interpolation in both computational cost and accuracy, which suggest they might be better suited for RTM-based LUT sampling.

The remainder of this paper is as follows. Section II gives a theoretical overview of the analyzed interpolation and emulation methods. Section III presents the materials and methods to study the performance of interpolation and emulation methods in terms of accuracy and computation time. This is followed by presenting the results in Section IV which are discussed in a broader context in Section V. Section VI concludes this paper.

II. INTERPOLATION AND EMULATION THEORY

In this section, we first present common interpolation methods (see Section II-A), and then address the emulation theory (see Section II-B).

A. Interpolation

Let us consider a D -dimensional input space \mathcal{X} from where we sample $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$ in which a K -dimensional object function $\mathbf{f}(\mathbf{x}; \lambda) = [f(\mathbf{x}; \lambda_1), \dots, f(\mathbf{x}; \lambda_K)] : \mathbb{R} \mapsto \mathbb{R}^K$ is evaluated. In the context of this paper, \mathcal{X} comprises the D input variables [e.g., leaf area index (LAI), aerosol optical thickness (AOT), visual zenith angle (VZA)] that control the behavior

of the function $\mathbf{f}(\mathbf{x}; \lambda)$, i.e., a water, canopy or atmospheric RTM. Here, λ represents the wavelengths in the K -dimensional output space.¹ An interpolation, $\hat{\mathbf{f}}(\mathbf{x})$ is, therefore, a technique used to approximate model simulations, $\mathbf{f}(\mathbf{x}) = \hat{\mathbf{f}}(\mathbf{x}) + \varepsilon$, based on the numerical analysis of an existing set of *nodes*, $\mathbf{f}_i = \mathbf{f}(\mathbf{x}_i)$, conforming a precomputed LUT. The concept of interpolation has been widely used in remote sensing applications, including retrieval of biophysical parameters [39] and atmospheric correction algorithms [26], [27]. The following nonexhaustive list gives an overview of commonly used interpolation techniques in remote sensing.

- 1) *Nearest-Neighbor*: This is the simplest method for interpolation, which is based on finding the closest LUT *node* \mathbf{x}_i to a query point \mathbf{x}_q (e.g., by minimizing their Euclidean distance) and associate their output variables, i.e., $\hat{\mathbf{f}}(\mathbf{x}_q) = \mathbf{f}(\mathbf{x}_i)$. This fast method is valid for both gridded and scattered LUTs.
- 2) *Inverse Distance Weighting (IDW)* [40]: Also known as *Shepard's method*, this method weights the n closest LUT nodes to the query point \mathbf{x}_q (1) by the inverse of the distance metric $d(\mathbf{x}_q, \mathbf{x}_i) : \mathcal{X} \mapsto \mathbb{R}^+$ (e.g., the Euclidean distance)

$$\hat{\mathbf{f}}(\mathbf{x}_q) = \frac{\sum_{i=1}^n \omega_i \mathbf{f}(\mathbf{x}_i)}{\sum_{i=1}^n \omega_i} \quad (1)$$

where $\omega_i = d(\mathbf{x}_q, \mathbf{x}_i)^{-p}$ and p (typically $p = 2$) is a tunable parameter known as *power parameter*. When p is large, this method produces the same results as the nearest-neighbor interpolation. The method is computationally cheap but it is affected by LUT nodes far from the query point. The modified Shepard's method [41] aims to reduce the effect of distant *grid points* by modifying the weights by

$$\omega_i = \left(\frac{R - d(\mathbf{x}_q, \mathbf{x}_i)}{R \cdot d(\mathbf{x}_q, \mathbf{x}_i)} \right)^p \quad (2)$$

where R is the maximum Euclidean distance to the n closest LUT nodes.

- 3) *Piece-Wise Linear*: This method is commonly used in remote sensing applications due to its balance between computation time and interpolation error [26], [42], [43]. The implementation of the linear interpolation is based on the Quickhull algorithm [44] for triangulations in multidimensional input spaces. For the scattered LUT input data, the piece-wise linear interpolation method is reduced to find the corresponding Delaunay simplex [45] (e.g., a triangle when $D = 2$) that encloses a query D -dimensional point \mathbf{x}_q [see (3) and Fig. 1]

$$\hat{\mathbf{f}}_i(\mathbf{x}_q) = \sum_{j=1}^{D+1} \omega_j \mathbf{f}(\mathbf{x}_j) \quad (3)$$

where ω_j are barycentric coordinates of \mathbf{x}_q with respect to the D -dimensional simplex (with $D + 1$ vertices) [46].

¹For sake of simplicity, the wavelength dependency is omitted in the formulation in this paper, i.e., $\mathbf{f}(\mathbf{x}; \lambda) \equiv \mathbf{f}(\mathbf{x})$.

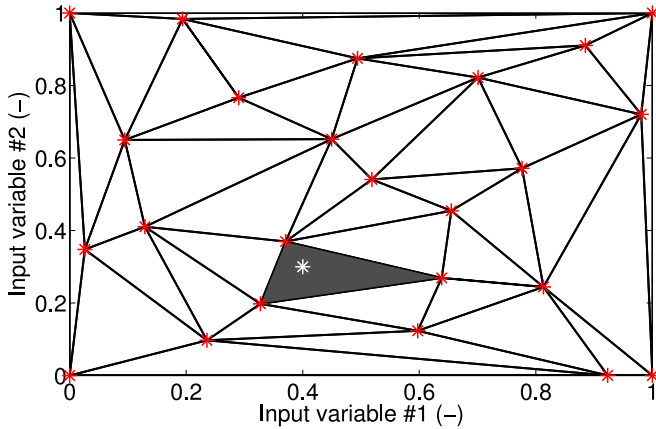


Fig. 1. Schematic representation of a two-dimensional interpolation of a query point \mathbf{x}_q (white $*$) after Delaunay triangulation (solid lines) of the scattered LUT nodes X_i ($*$).

Notice that, though similar, the IDW with parameters $n = D + 1$ and $p = 1$ is not strictly the same as linear interpolation since IDW uses Euclidean distances instead of the barycentric coordinates in linear interpolation. Since $\mathbf{f}(\mathbf{x})$ is a K -dimensional function, the result of the interpolation is also K -dimensional.

In scattered LUTs, the underlying Delaunay triangulation is computationally expensive in high dimensional input spaces (typically $D > 6$) and is also limited by its intensive memory consumption [44], [47].

Other existing advanced interpolation methods (e.g., Sibson's interpolation [46], [48] and piece-wise cubic splines [49]) were not considered in the analysis due to their even more intensive memory consumption in high-dimensional input spaces.

B. Emulation

Emulation is a statistical learning technique used to estimate model simulations when the model under investigation is too computationally costly to be run many times [29]. The concept of developing emulators have already been applied in the last few decades in the climate and environmental modeling communities [19], [50]–[56]. The basic idea is that an emulator uses a limited number of simulator runs, i.e., input-output pairs (corresponding to training samples), to train a machine learning regression algorithm in order to infer the values of the complex simulator output given a yet-unseen input configuration. These training data pairs should ideally cover the multidimensional input parameter space using a space-filling sampling algorithm, e.g., Latin hypercube sampling [57].

As with the LUT interpolation, once the emulator is built, it is not necessary to perform any additional runs of the model; the emulator computes the output that is otherwise generated by the simulator [29]. Accordingly, emulators are statistical models that can generalize the input–output relations from a subset of examples generated by RTMs to unseen data. Note that building an emulator is essentially nothing more than building an advanced regression model as typically done for biophysical parameter retrieval applications [see pioneering works using

neural networks (NNs) [58], [59] and also more recent statistical methods [6], [60], [61], but in reversed order: whereas a retrieval model converts input spectral data (e.g., reflectance) into one or more output biophysical variables, an emulator converts input biophysical variables into output spectral data.

When it comes to emulating RTM spectral outputs, however, the challenge lies in delivering a full spectrum, i.e., predicting contiguous spectral bands. This is an additional difficulty compared to traditional interpolation methods or standard emulators that only deliver one output [62]. It bears the consequence that the machine learning methods should be able to generate multiple outputs to be able reconstructing a full spectral profile. This is not a trivial task. For instance, the full, contiguous spectral profile between 400 and 2500 nm consists of over 2000 bands when binned to 1 nm resolution. Not all regression models are able to deal with high-dimensional outputs. Only some of them can obtain multioutput models. For instance, with NNs it is possible to train multioutput models. However, training a complex multioutput statistical model with the capability to generate so many output bands would take considerable computational time and would probably incur in a certain risk of overfitting because of model overrepresentation. A workaround solution has to be developed that enables the regression algorithms to cope with large, spectroscopy datasets. An efficient solution is to take advantage of the so-called *curse of spectral redundancy*, i.e., the Hughes phenomenon [63]. Since spectroscopy data typically shows a great deal of collinearity, it implies that such data can be converted to a lower-dimensional space through dimensionality reduction (DR) techniques. Accordingly, spectroscopy data can be converted into components, which are only a fraction of the original amount of bands, and implies that the multioutput problem is greatly reduced to a number of components that preserve the spectral information content. Afterward the components are then again reconstructed to spectral data. In this paper, we first apply a principal component analysis (PCA) [64] to the spectral data in order to reduce it to a given number of features (components). This step greatly reduces the number of dimensions while keeping 99% of the spectra variance. Through DR, the problem is better conditioned and allows us to either train multioutput or single-output models on this reduced set of components [30]–[32], [55], [65]. As the models are trained to predict on the reduced set of components, the final step of the process is to project back the predictions to the original spectra space by applying the inverse PCA.

C. Machine Learning Regression Algorithms

Two steps are required to enable approximating an RTM through emulation. The first step involves building a statistically-based representation (i.e., an emulator) of the RTM using statistical learning from a set of training data points derived from runs of the actual model under study (LUT nodes in the context of interpolation). The second step uses the emulator previously built in the first step to compute the output in the LUT input parameter space that would otherwise have to be generated by the original RTM [29]. Based on the literature review above and earlier emulation evaluation studies

[31], [32], [65], the following three machine learning methods potentially serve as powerful methods to function as accurate emulators, being: First, Gaussian processes regression (GPR); second, kernel ridge regression (KRR); and third, NNs.

We selected these three methods as representative the state-of-the-art machine learning families for regression. KRR generalizes linear regression via kernel functions. The GPR is essentially the probabilistic version of the KRR, and has been widely used for biogeophysical parameter and emulation [36], [66], [67]. NNs are standard approximation tools in statistics and artificial intelligence, and are currently revived through the popular adoption of deep learning models [68]. We explore all these techniques for the sake of a complete benchmarking of standard methods available. These methods are briefly outlined later.

Kernel methods in the machine learning owe their name to the use of kernel functions [69]–[71]. These functions quantify similarities between input samples of a dataset. Similarity reproduces a linear dot product (scalar) computed in a possibly higher dimensional feature space, yet without ever computing the data location in the feature space. The following two methods are gaining increasing attention: the GPR generalize Gaussian probability distributions in function spaces [72], and KRR, which perform least squares regression in feature spaces [73]. The expressions defining the weights and the predictions obtained by GPR and KRR are the same, but interestingly these expressions are obtained following different approaches. GPR follow a probabilistic approach (see [72]), whereas KRR implement a discriminative approach for regression and function approximation. In both cases, the prediction and the predictive variance of the model for new samples are given by

$$\hat{\mathbf{f}}(\mathbf{x}_q) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x}_q) \quad (4)$$

$$\mathbb{V}[\hat{\mathbf{f}}(\mathbf{x}_q)] = k(\mathbf{x}_q, \mathbf{x}_q) - \mathbf{k}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_* \quad (5)$$

where $k(\cdot, \cdot)$ is a covariance (or kernel function), \mathbf{k}_* is the vector of covariances between the query point, \mathbf{x}_q , and the n or training points, and σ_n^2 accounts for the noise in the training samples. As one can see, the prediction is obtained as a linear combination of weighted kernel (covariance) functions, the optimal weights given by $\alpha = (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{f}(\mathbf{x})$. Many different functions can be used as kernels for both GPR and KRR. In this paper, we used a standard Gaussian radial basis function kernel for KRR, which has a single length hyperparameter for all input dimensions, and the automatic relevance determination squared exponential kernel for GPR, which has a separate length hyperparameter for each input dimension. For KRR, these hyperparameters are tuned through standard cross-validation techniques are used to choose the best hyperparameters. For GPR, stochastic gradient descent algorithms maximizing the marginal log-likelihood are employed, which allow us to optimize a large number of hyperparameters (compared to KRR) in a computational effective way.

NNs are essentially fully connected layered structures of artificial neurons [74]. An NN is a (potentially fully) con-

nected structure of neurons organized in layers. Neurons of different layers are interconnected with the corresponding links (weights). The output on the final layer of the NN, and thus the prediction, is given by

$$\hat{\mathbf{f}}(\mathbf{x}_*) = g \left(\sum_{k=1}^n w_{jk}^l x_k^{l-1} + b_k^l \right) \quad (6)$$

where w_{jk}^l and b_k^l are the weights and bias at the l th layer, respectively, x_k^{l-1} is the input vector at the $l-1$ th layer, and g is an activation function, which at the output layer and for regression problems could be the identity function. Training an NN implies selecting a structure (number of hidden layers and nodes *per* layer), initialize the weights, shape of the nonlinear activation function, learning rate, and regularization parameters to prevent overfitting [75]. The selection of a training algorithm and the loss function both have an impact on the final model. In this paper, we used the standard multilayer perceptron, which is a fully-connected network. We selected just one hidden layer of neurons. We optimized the NN structure using the Levenberg–Marquardt learning algorithm with a squared loss function.

One can note the similarity between the prediction functions used in the emulators [see (4) and (6)] with those used for interpolation (see Section II-A). In fact, the theoretical relation between both approaches was extensively discussed back in 1970 in the context of splines and GPR in [76]. Essentially, machine learning emulators perform regression and, hence, are more flexible functions for fitting than interpolation, as the solution is not forced to pass through the observed points. On the downside, the emulation approach may be hampered by an adequate estimation of the hyperparameters (i.e., regularization). When a good estimate of the hyperparameters is achieved emulation should obtain equal or better results than interpolation, otherwise the regression may incur in a certain risk of overfitting.

III. MATERIALS AND METHODS

In this section, we will start by giving an overview the software used to generate the synthetic datasets used to assess the performance of the interpolation/emulation methods (see Sections III-A and III-B). We will continue by describing these datasets (see Section III-C) and finish by explaining the error metrics used to evaluate the performance of the interpolation/emulation methods (see Section III-D).

A. Automated Radiative Transfer Models Operator (ARTMO) and Atmospheric Look-Up Table Generator (ALG) Toolboxes

This study was conducted within two in-house developed graphical user interface software packages named ARTMO [77] and ALG [78]. Both software packages facilitate the usage of a suite of leaf, canopy and atmosphere RTMs including, among others, PROSAIL (i.e., the leaf model PROSPECT coupled with the canopy model SAIL [79]) and MODTRAN5. As a novelty, the latest ARTMO version (v. 3.24) is coupled with ALG (v. 1.2), which allows generating large multidimensional LUTs of TOA radiance data for Lambertian surfaces.

ARTMO also embodies a set of retrieval toolboxes, and recently an “Emulator toolbox” was added [65]. In the Emulator toolbox, several of those MLRAs can be trained by RTM-generated LUTs, whereby biophysical variables are used as input in the regression model, and spectral data is generated as an output. In addition, ALG includes a function that allows various methods of interpolating gridded and scattered LUTs (i.e., nearest neighbor, piece-wise linear/splines, IDW). The ARTMO and ALG packages are developed in MATLAB and can be freely downloaded from <http://ipl.uv.es/artmo/>.

B. Description of Simulated Datasets

1) *PROSPECT-4*: The leaf optical model PROSPECT-4 [11] calculates leaf reflectance and transmittance as a function of four biochemistry and anatomical variables: leaf structure (N), equivalent water thickness (Cw), chlorophyll content (Cab), and dry matter content (Cm). PROSPECT-4 simulates directional reflectance and transmittance over the solar spectrum from 400 to 2500 nm at the fine spectral resolution of 1 nm.

2) *SAIL*: At the canopy scale, SAIL [12] approximates the RT equation through two direct fluxes (incident solar flux and radiance in the viewing direction) and two diffuse fluxes (upward and downward hemispherical flux) [80]. SAIL input variables consist of LAI, leaf angle distribution (LAD), ratio of diffuse and direct radiation (skyl), soil coefficient (soil coeff.), hot spot and sun-target-sensor geometry, i.e., solar/view zenith angle and relative azimuth angle (SZA, VZA and RAA, respectively). Spectral input consists of leaf reflectance and transmittance spectra and a soil reflectance spectrum. The leaf optical properties can come from a leaf RTM such as PROSPECT, which results in the leaf-canopy model PROSAIL [3]. PROSAIL allows analyzing the impact of leaf biochemical variables on the hemispherical and bidirectional TOC reflectance.

3) *MODTRAN5*: At the atmosphere scale, MODTRAN5 [21], the moderate resolution transmittance code, is one of the most widely used radiative transfer codes in the atmospheric community due to its accurate simulation of the coupled absorption and scattering effects [81], [82]. MODTRAN solves the RT equation in a multilayered spherically symmetric atmosphere by including the effects of molecular and particulate absorption/emission and scattering, surface reflections and emission, solar/lunar illumination, and spherical refraction.

C. Experimental Setup

Here, we outline the experimental setup for running the interpolation and emulation experiments. For both PROSAIL and MODTRAN RTMs, LUTs were generated by means of Latin hypercube sampling (LHS) within the RTM variable space with minimum and maximum boundaries as given in Tables I and II. The selected input variables were chosen given their influence in both the entire spectra (e.g., Aerosol optical thickness, Ångström exponent) and in specific absorption bands (e.g., Chlorophyll absorption, water vapour, and ozone). An LHS of training data is preferred, as LHS covers the full parameter space, and thus, in principle, assures that the developed emulator/interpolation

TABLE I
RANGE OF VEGETATION INPUT VARIABLES FOR THE PROSAIL LUTS
ACCORDING TO LATIN HYPERCUBE SAMPLING

Model variables		Units	Minimum	Maximum
Leaf variables (PROSPECT-4)				
N	Leaf structure index	unitless	1.3	2.5
Cw	Leaf water content	[cm]	0.002	0.05
Cab	Leaf chlorophyll content	[$\mu\text{g}/\text{cm}^2$]	1	70
Cm	Leaf dry matter content	[g/cm^2]	0.002	0.05
Canopy variables (SAIL)				
LAI	Leaf area index	[m^2/m^2]	0.1	7
LAD	Leaf angle distribution	[$^\circ$]	0	90

SAIL fixed variables: hot spot: 0.01; solar zenith angle: 30° ; observer zenith angle: 0° ; azimuth angle: 0° .

TABLE II
RANGE OF ATMOSPHERIC INPUT VARIABLES FOR THE MODTRAN LUTS
ACCORDING TO LATIN HYPERCUBE SAMPLING

Model variables		Units	Minimum	Maximum
O3C	O ₃ column concentration	[amt-cm]	0.2	0.45
CWV	Columnar Water Vapour	scale-factor	1	4
AOT	Aerosol Optical Thickness	unitless	0.05	0.4
G	Asymmetry parameter	unitless	0.65	0.99
α	Ångström exponent	unitless	1	2
SSA	Single Scattering Albedo	unitless	0.75	1

MODTRAN fixed geometric variables: solar zenith angle: 55° ; observer zenith angle: 0° ; azimuth angle: 0° . Remaining MODTRAN parameters were set to their default values.

will be able to reconstruct correct spectral output for any possible combination of input variables. For both the canopy and atmospheric RTMs, three sizes of LUTs were created given the same LUT boundaries: 500, 2000, and 5000. While the most dense LUT (5000) was used as a reference LUT to evaluate the performances of the emulation and interpolation algorithms, the first two LUTS (500 and 2000) were simulated to actually run the emulation and interpolation techniques for different LUTs sizes. Additionally, the 64 vertex of the input variable space (i.e., where the input variables get the minimum/maximum values) were added to these two LUTs. The addition of these vertex enables consistent functioning of all tested interpolation techniques, i.e., that the input variable space is bounded and no extrapolation is performed.

The MODTRAN LUTs consist on TOA radiance spectra constructed according to (7) under the Lambertian assumption

$$L_{\text{TOA}} = L_0 + \frac{(E_{\text{dir}}\mu_s + E_{\text{dif}})(T_{\text{dif}} + T_{\text{dir}})\rho}{\pi(1 - S\rho)} \quad (7)$$

where L_0 is the path radiance, $E_{\text{dir/dif}}$ are the direct/diffuse at-surface solar irradiance, $T_{\text{dir/dif}}$ are the surface-to-sensor direct/diffuse atmospheric transmittance, S is the spherical albedo, μ_s is the cosine of SZA, and ρ is the Lambertian surface reflectance (in our case we used the conifer trees surface reflectance from ASTER spectral library [83]). The atmospheric transfer functions are derived after applying the MODTRAN interrogation technique described in [26].

For the emulation approach, each LUT was used to develop and evaluate the different statistical models.

The role of number of components has been systematically studied before [32]. The selection of 10 and 20 PCA components (i.e., $\sim 100\%$ explained variance) was found an acceptable trade-

off between accuracy and processing time. Better reconstruction of the spectral profiles can be achieved with additional components, but at expenses of slower processing times. Further, since emulators only produce an approximation of the original model, it is important to realize that such an approximation introduces a source of uncertainty referred to as “code uncertainty” associated to the emulator [29]. Therefore, validation of the generated model is an important step in order to quantify the emulator’s degree of accuracy. To test the accuracy of the 500- and 2000-LUT emulators, part of the original data is kept aside as validation dataset. Various training/validation sampling design strategies are possible with the “Emulator toolbox.” Because of the deterministic nature of RTM data, an initial cross-validation sampling testing led to similar accuracies as one-time validation. To speed up the processing time [31], a single data split was, therefore, applied using 70% samples for training and the remaining 30% for validation.

D. Validation

In order to show the differences between the RTM outputs and the approximation inferred by interpolation or emulation techniques, some goodness-of-fit statistics as a function of wavelength are calculated against the $n = 5000$ references LUTs as generated by the RTMs. The root-mean-square error (RMSE) and the normalized RMSE (NRMSE) [%] [see (8) and (9)] are calculated, both per wavelength and then averaged over all wavelengths (λ)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n [\mathbf{f}(\mathbf{x}_i) - \hat{\mathbf{f}}(\mathbf{x}_i)]^2} \quad (8)$$

$$\text{NRMSE} = \frac{100 \cdot \text{RMSE}}{\mathbf{f}_{\max} - \mathbf{f}_{\min}} \quad (9)$$

where \mathbf{f}_{\max} and \mathbf{f}_{\min} are, respectively, the maximum and minimum values of the n spectra in the reference dataset. A closer inspection will be given to the most interesting results by plotting the histogram of the relative residuals (ε_i , in absolute terms and expressed in %)

$$\varepsilon_i = 100 \frac{|\mathbf{f}(\mathbf{x}_i) - \hat{\mathbf{f}}(\mathbf{x}_i)|}{\mathbf{f}(\mathbf{x}_i)} \quad (10)$$

Specifically, the average relative error and the percentiles 2.5%, 16%, 84%, and 95.5% will be plotted as function of wavelength.

The processing time of executing the emulator/interpolation method on the reference dataset has also been tracked. These calculations were performed in a i7-4710MP CPU at 2.5 GHz with 16 GB of RAM and 64-bits operating system.

IV. RESULTS

In this section, we will show the results of applying the emulator and interpolation methods on the described canopy and top-of-atmosphere datasets. In Section IV-A, we will show an overview of the performance of interpolation and emulation methods in terms of accuracy and computation time. In Section IV-B, we will inspect in greater detail the error his-

TABLE III
PROSAIL INTERPOLATION AND EMULATORS VALIDATION RESULTS AGAINST 5000 LUT REFERENCE DATASET (RMSE $_{\lambda}$, NRMSE $_{\lambda}$) AND PROCESSING TIME (S: SECONDS)

Method	RMSE $_{\lambda}$		NRMSE $_{\lambda}$ (%)		CPU (s)	
	500	2000	500	2000	500	2000
Interpolation:						
- Nearest	0.051	0.042	11.60	9.61	0.2	0.7
- Linear	0.042	0.030	9.99	7.05	62	171
- IDW	0.039	0.032	8.86	7.44	0.5	1.2
Emulation 10PCA:						
- GPR	0.005	0.003	1.23	0.68	0.7	2.2
- KRR	0.015	0.007	3.56	1.67	0.1	0.2
- NN	0.024	0.022	5.33	5.00	0.2	0.2
Emulation 20PCA:						
- GPR	0.005	0.003	1.21	0.64	0.9	4.2
- KRR	0.015	0.007	3.54	1.67	0.3	0.3
- NN	0.021	0.022	4.74	5.04	0.2	0.2

tograms for the best performing interpolation and emulation methods.

A. Interpolation Versus Emulation Comparison

For both PROSAIL and MODTRAN outputs four scenarios are evaluated: training/interpolating with 500 and 2000 samples. The emulation approach is additionally tested with entering 10 or 20 components in the regression algorithm. All approaches are validated against the reference 5000 samples’ LUTs.

Starting with the PROSAIL analysis, validation results and processing time is given in Table III. NRMSE results along the spectral range for the four scenarios are shown in Fig. 2. Inspection of these four graphs suggest the following. Each of the four scenarios show approximately the same patterns, with expected higher NRMSE errors in spectral channels with lower reflectance values (e.g., bottom of Chlorophyll absorption at 680 nm and inside the water absorptions at 1440 nm and 1900 nm). The three emulation methods clearly outperform the three interpolation methods in reproducing LUT reflectance spectra. The GPR is best able to reconstruct spectra with high accuracies (i.e., low NRMSE errors). The KRR is second best performing, while NN performs still better than the interpolation techniques but no longer with a substantial gain in accuracy. Among the interpolation methods, linear and IDW achieve similar accuracy, particularly when using the 2000 samples LUT. Only in the near-infrared plateau (i.e., 720–1300 nm) linear interpolation obtain the lowest NRMSE errors among the interpolation techniques, similar to those obtained with NN emulation. Thereby, results improved when more input data is involved, i.e., when the statistical models are trained with more samples, or when a denser LUT is used for interpolation. This is clearly notable when comparing the results of 2000 samples with that those of 500. A decrease in errors is especially noticeable for KRR, but also the interpolation methods lower errors with a few percents. The superiority of emulation methods can perhaps be better appreciated when considering Table III: GPR trained by a 2000-LUT yielded RMSE $_{\lambda}$ on the order of ten times lower than the best interpolation method.

PROSAIL emulation results can further improved when more components are entered in the statistical learning, as then in

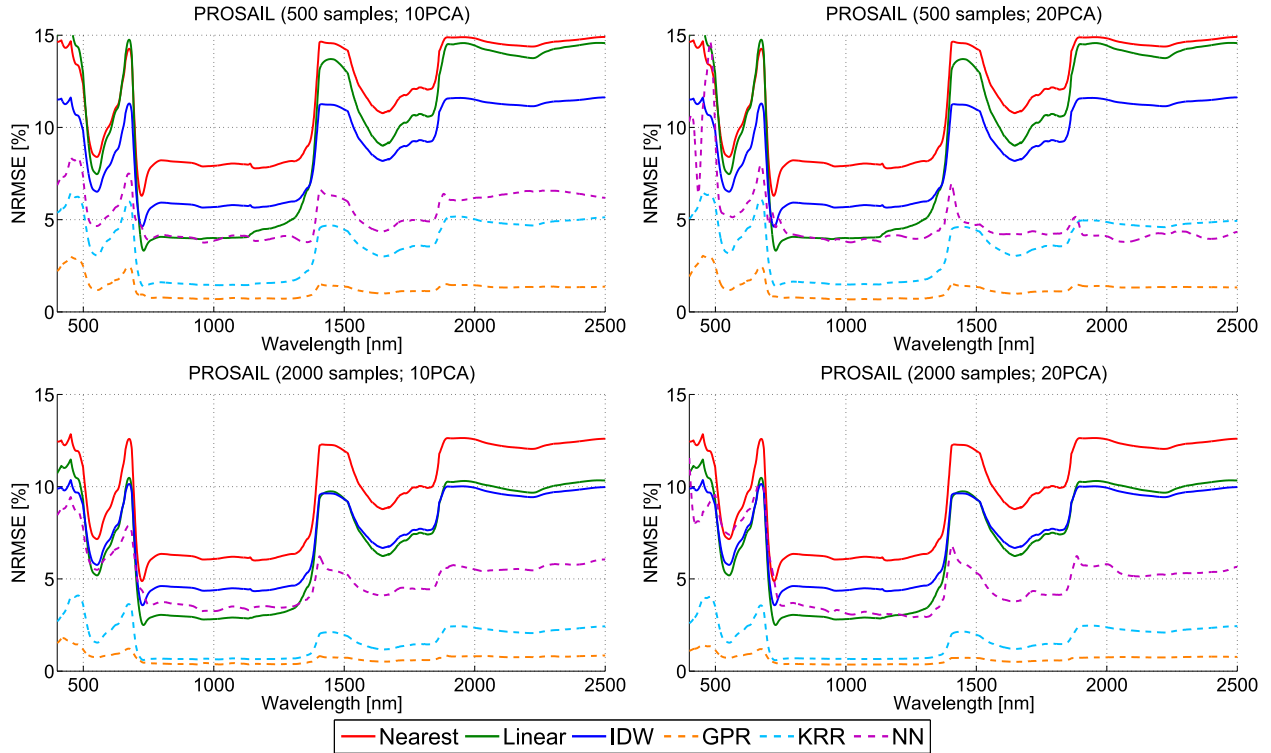


Fig. 2. PROSAIL interpolation versus emulation results. Note that the number of PCA components refers only to the emulator methods since no DR is applied in interpolation.

principle more variability is preserved. However, these improvements were not obvious in our results: when doubling the number of components from 10 to 20 hardly differences were observed for KRR and GPR. This is especially the case for the 2000 training LUT: Table III gives the same $RMSE_{\lambda}$ results. Hence, this suggests that about ten components are more than enough to preserve a maximal amount of information. NN appears more affected by the number of components in case of the 500 training samples: clear improvements can be observed from 1500 nm onward. Conversely, in the visible part errors increased, which implies that the gain of adding more components is not systematic. In case of trained with 2000 samples then doubling the components did not influence at all.

When subsequently also considering processing time (see Table III), then the emulation methods become even more attractive. Although the interpolation methods nearest neighbor and IDW are very fast (below 1% of the slowest, linear interpolation, method), they are not the most accurate: especially nearest neighbor is fastest but also the poorest performing. On the contrary, the emulation methods are not only accurate, but are also very fast. GPR processes the output spectra with a speed that is on the order of these interpolation methods. However, the GPR is affected by the training size and number of components, which slows down somewhat the processing. Yet, even for the 2000 LUT and including 20 components the processing of 5000 output spectra took only a few seconds, i.e., 2.4% of the time spent by linear interpolation. NN and KRR deliver spectral output still several times faster, in a fraction of a second, and this regardless of the training size. Hence, when a tradeoff be-

TABLE IV
MODTRAN INTERPOLATION AND EMULATORS VALIDATION RESULTS AGAINST 5000 LUT REFERENCE DATASET ($RMSE_{\lambda}$, $NRMSE_{\lambda}$) AND PROCESSING TIME (S: SECONDS)

Method	$RMSE_{\lambda}$		$NRMSE_{\lambda}$ (%)		CPU (s)	
	500	2000	500	2000	500	2000
LUT training size:						
Interpolation:						
- Nearest	1.160	0.896	7.58	5.83	0.3	0.8
- Linear	0.386	0.265	2.84	1.98	68	183
- IDW	0.837	0.630	5.47	4.12	0.5	1.3
Emulation 10PCA:						
- GPR	0.037	0.031	0.65	0.59	0.5	2.0
- KRR	0.100	0.054	1.01	0.73	0.1	0.2
- NN	0.488	0.406	4.13	3.62	0.1	0.1
Emulation 20PCA:						
- GPR	0.029	0.022	0.43	0.37	1.2	4.7
- KRR	0.094	0.051	0.84	0.56	0.1	0.2
- NN	0.487	0.466	4.83	4.40	0.1	0.1

tween accuracy and processing speed is to be made, given that NN delivers poorer accuracies, then KRR tends to become an attractive option. In all cases KRR emulated the 5000 spectra a factor 100 faster than linear interpolation (0.3 s), and this with second-best accuracies.

The same analysis has been repeated with the MODTRAN LUTs (see Table IV and Fig. 3). $NRMSE$ results are now plotted in logarithmic scale in order to better visualize differences between the various interpolation/emulation methods and the wide range of error values between inside and outside atmospheric absorption bands. Similar patterns as the PROSAIL results are obtained, yet some differences should be remarked. GPR and sec-

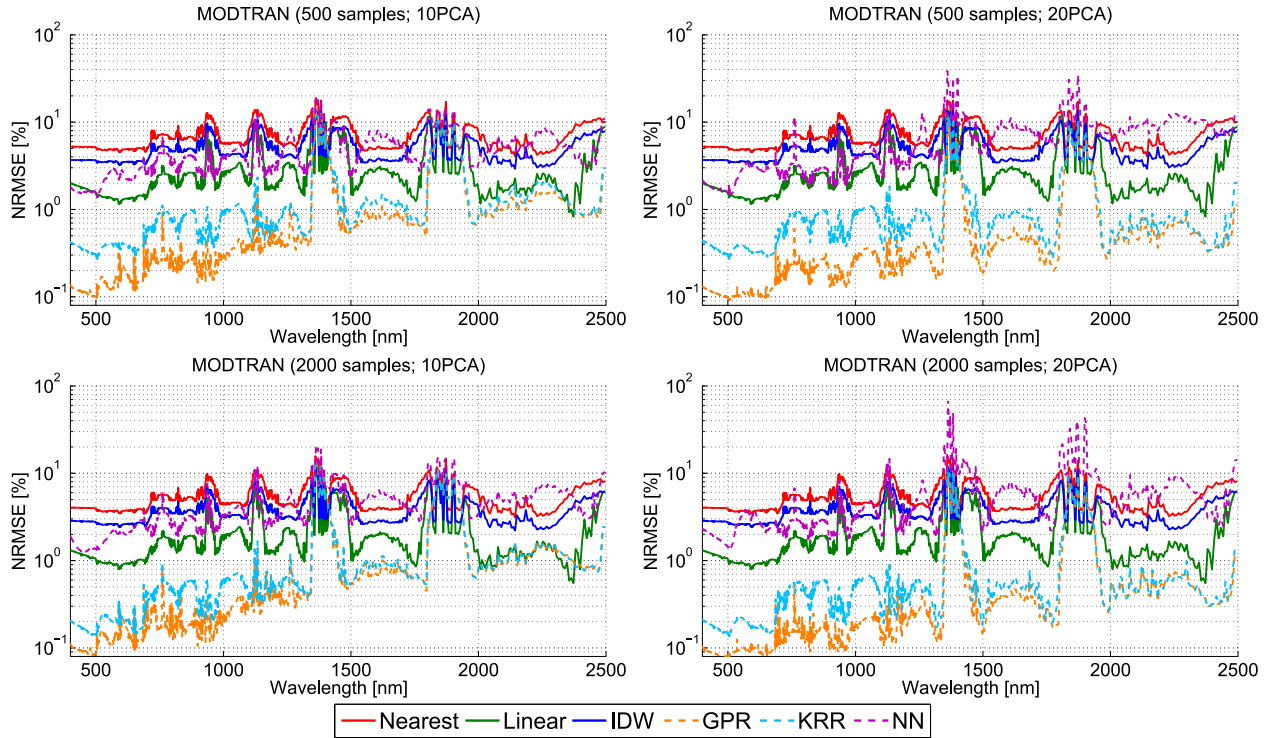


Fig. 3. MODTRAN interpolation versus emulation results. Note that the number of PCA components refers only to the emulator methods since no DR is applied in interpolation.

and KRR are again clearly top performing LUT parameter sampling methods. Table IV indicates that KRR and especially GPR yielded $RMSE_{\lambda}$ results more than ten times lower than the best performing interpolation method. However, NN performs now on the order of the interpolation methods. Regarding the interpolation methods, linear interpolation now systematically outperformed the other two methods, but still errors are nearly one order of magnitude higher than the GPR and KRR emulators. Moving from 500 to 2000 training samples did not lead to significant improvements. Table IV suggests that only for KRR a substantial gain in accuracy was achieved. The same holds for adding more components into the emulators: although some small improvements can be obtained with more components, e.g., as is noticeable for GPR, overall the gain in accuracy is modest.

Also regarding processing time similar trends emerged as for PROSAIL: all three emulation methods produced the spectral output very fast, with NN and KRR delivering the 5000 MODTRAN-like spectra in a fraction of a second (a factor $<1\%$ when compared against linear interpolation). GPR suffered somewhat from adding more samples and components, leading to a slightly slower emulation method in case of 2000 LUT and trained with 20 components: the output spectra is again produced in a few seconds.

B. Closer Inspection of Best Performing Results

Having observed the general trends of the interpolation and emulation methods, in this section, we will inspect a few methods in more detail. Specifically, the histograms of the relative residuals (in absolute terms) for the best performing

interpolation and emulation methods, i.e., linear and GPR, are plotted in Figs. 4 and 5. Thereby, the linear interpolation method is shown as obtained with a 2000-LUT, whereas the GPR is shown as trained with only a 500-LUT and 10 PCA components. Interestingly, although the emulator method is not presented in its optimized configuration, already a substantial gain in accuracy as compared to the optimized linear interpolation method is achieved.

To fully appreciate the predictive power of the emulator methods, we start by analyzing the results on the PROSAIL residuals in Fig. 4. We can observe how the GPR emulator obtains relative errors that, on average, are a factor 5–10 lower than those obtained with the linear interpolation (see mean values on the solid line). These error differences between the GPR emulator and linear interpolation methods are still maintained on both the lower and higher part of the histogram (see 2.5% and 84% percentiles, where the reconstruction errors are lower/higher, respectively).

We continue by analyzing the results on the MODTRAN residuals in Fig. 5. As previously observed in NRMSE values (see Fig. 3 and Table IV), the reconstruction errors with GPR emulators obtains the best performance. The residual errors are in this case a factor 2–10 lower than using linear interpolation for both the lowest and highest errors in the histogram and in most part of the spectrum (<1800 nm).

V. DISCUSSION

Advanced RTMs are widely used in various remote sensing applications, such as atmospheric correction (e.g., [84]),

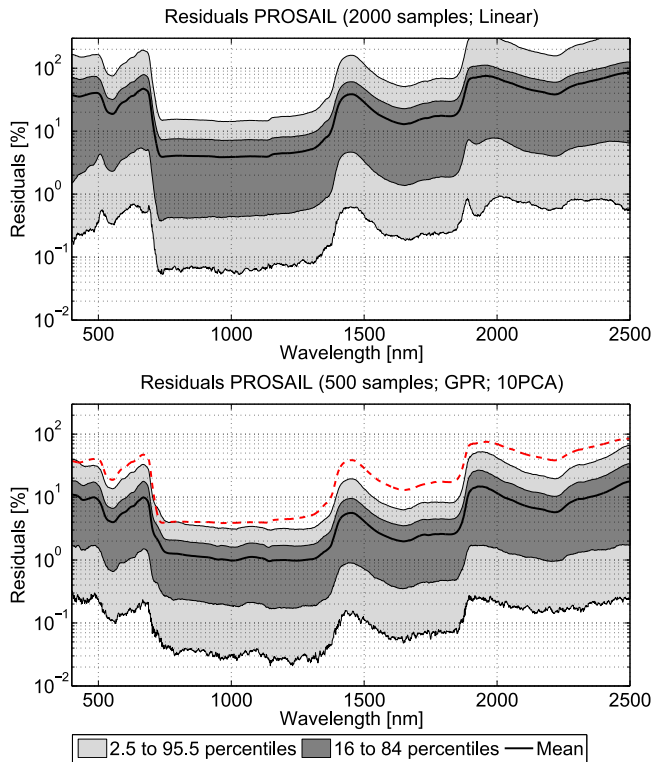


Fig. 4. Histogram statistics of PROSAIL relative residuals (in absolute terms)(%) for 2000-LUT interpolation linear (top) and 500-LUT GPR (bottom). To ease the comparison, the mean residual for linear interpolation is added on top of the GPR residuals (red dashed line).

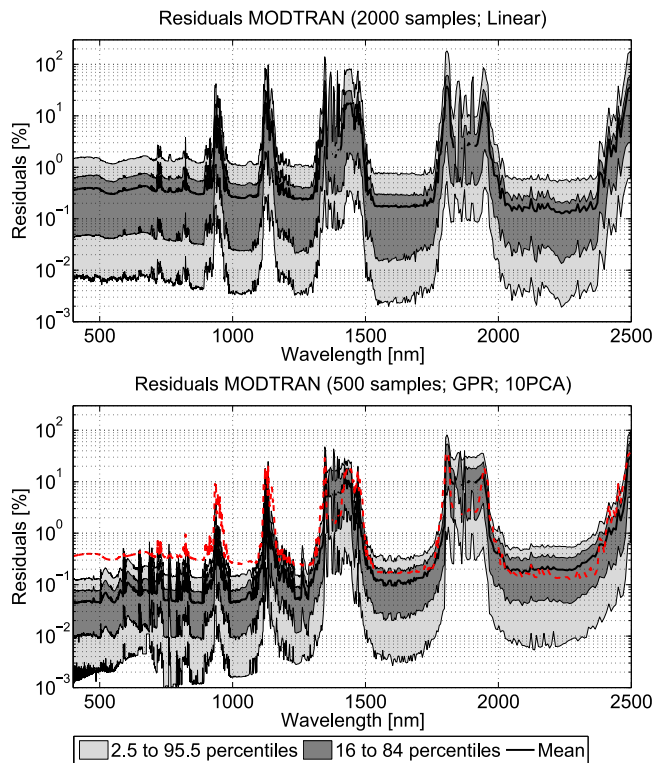


Fig. 5. Histogram statistics of MODTRAN relative residuals (in absolute terms)(%) for 2000-LUT interpolation linear (top) and 500-LUT GPR (bottom). To ease the comparison, the mean residual for linear interpolation is added on top of the GPR residuals (red dashed line).

inversion of vegetation properties (see [6] for a review), sensitivity analysis [10], and scene generation [7]. Because advanced RTMs take long processing time, the LUT interpolation is commonly used to sample the input parameter space and to infer an approximation of the RTM outputs [26]. Although various interpolation techniques are standard practice in remote sensing applications and computer vision, in this paper, we challenged these techniques by comparing them against statistical learning methods, i.e., *emulation*. The basic principle of emulation is using a sparse LUT to train machine learning methods so that the trained statistical model is able to reproduce the spectral output given unseen parameter combinations. According to this principle, the emulation technique can be considered to function similarly as interpolation methods, but based on statistical learning.

To ascertain the predictive power of both interpolation and emulation, two experiments were conducted: one for surface reflectance as generated by PROSAIL, and another for TOA radiance data as generated by MODTRAN. For both experiments the interpolation and emulation methods were tested against a common reference dataset of 5000 simulations. Despite the TOA radiance data being much more irregular and spiky (due to narrow atmospheric absorption regions) than surface reflectance, common trends were observed in the majority of cases. They are summarized as follows.

- 1) In all experiments, the three tested emulation methods produced substantially more accurately spectral outputs than the three tested interpolation techniques. Particularly, GPR was by far the most accurate emulation method with errors on the order of 5–10 times lower than the linear interpolation method and at a fraction of the time with respect linear interpolation. The KRR was the second most accurate emulation method, with reconstruction errors 2–5 times lower than the best performing interpolation method. The GPR is thus clearly the preferred method given accuracy and fast processing. At the same time, the KRR run still much faster than GPR, i.e., producing 5000 spectra in 0.3 s, which is faster than any interpolation method behind while reaching superior accuracies.
- 2) Regarding the emulation methods, the GPR and KRR emulate output spectra not only perform more accurately, but also tend to be more stable than the NN (e.g. Fig. 2, in the visible part). Having more parameters than KRR/GPR to adjust, training the NN emulator can be more complicated, and thus requires a larger number of training samples to avoid overfitting and to obtain a good prediction. In addition, an increasing number of components can make the problem harder in terms of adjusting the NNs parameters. Conversely, for GPR and KRR clearly some little extra gain can be achieved by training with more samples or by adding more PCA components in the regression model. However, this slows down processing time, especially in case of GPR (see also [32] for a discussion on GPR emulation).
- 3) When comparing the PROSAIL emulation results with those of MODTRAN, e.g., as in the histogram statistics (Figs. 4 and 5), it is noteworthy that the spectrally

smoother TOC reflectance is reconstructed with lower accuracy than the more irregular TOA radiance. Although the parameter space of both LUTs consists of six variables, the discrepancy can be explained that the six PROSAIL variables exert more influence, in relative terms, on the reflectance data than the six MODTRAN variables on the TOA radiance data, implying more variability to construct the reflectance spectra. This has been observed before with a global sensitivity analysis [31].

- 4) Regarding the interpolation methods, while performing substantially poorer than emulation, the linear interpolation is the most accurate method. This is particularly the case when increasing the LUT size. However, it is also the most computationally intensive interpolation method, i.e., on the order of few minutes for 5000 spectra. This is mostly due to the exponential increase of memory usage in construction of the implicit Delaunay triangulation [44].

The superiority of KRR and especially GPR emulation is most noteworthy. The strength of the GPR statistical learning was already demonstrated in previous studies with application of machine learning algorithms in various fields [66], [85], [86], yet in fact any machine learning regression algorithm can function as emulator. Moreover, since the accuracy and run-time of these methods depend on the size of training LUT and number of components, these methods can be further optimized in view of balancing between accuracy and processing speed. For instance, it is likely that the GPR can still deliver excellent accuracies with a smaller training LUT or less components, which would imply a faster processing. As addressed before [32], it requires some iterations to deduce an optimized accuracy-speed tradeoff.

Hereby, another point to be remarked is that emulation requires the additional effort of a training step and validation of the model. This implies that a sparse LUT is always required to train an emulator, i.e., to finding the best hyperparameters that optimize its performance. Two issues have an impact on finding these optimal parameters: the method for tuning the hyperparameters and the size of the training dataset. Regarding the first issue, and in the specific case of the implemented GPR, the maximum log-likelihood method was adopted to find the best set of hyperparameters. Nevertheless, other alternative procedures are often employed with similar success and adoption. Examples include random sampling [87], the Nelder–Mead method (aka downhill simplex) [88], Bayesian optimization [88]–[91], and many flavors of derivative-free optimization approaches, such as stochastic local search, simulated annealing, or evolutionary computation [92], [93]. Other approaches consider Monte Carlo methods [94]–[96], which search in a portion of the space according to the posterior distribution of the hyperparameters given the observed data. With respect to the training dataset, we found in our examples that typically about 500 samples according Latin hypercube sampling should suffice. The training and model validation step requires some additional processing time as compared to interpolation. In case of KRR this is in the order of seconds, but NN and GPR that can take longer depending on the complexity of training setup (number of samples and components). Nevertheless, the training phase is to be done only one time; the generated emulator model can afterward replace

the LUT interpolation in the processing chain. This approach of replacing an LUT interpolation by an emulator may lead not only to a more accurate processing, but likely also faster and less computationally demanding.

With respect to interpolation methods, in this paper, we focused on exploring the accuracy of the most widely used methods working on scattered LUT data. From the considered methods, the linear interpolation achieve the higher accuracy. However, this method is typically limited to low dimensions (<6) of the input space due to its exponential demands of memory usage and computation time. Other more advanced interpolation methods (not studied here) exist for gridded LUT data. Among them, piece-wise cubic splines interpolation [49] can achieve high accuracies, but at expenses of an increase of computation time with respect linear interpolation. Also, cubic splines require gridded LUTs with at least four points in each dimension (e.g., 4096 LUT samples for six dimensions), which implies both an increase of memory usage and computation time to generate and interpolate the resulting LUTs. Sibson method is another advanced and accurate interpolation algorithm [?] that can have very fast implementations for low-dimension spaces (see, e.g., [48]). However, the extension of Sibson interpolation in high-dimensional input variable spaces (>6) is likely not to be effective due to its exponential growth in memory consumption. In turn, the emulation approach only requires a small LUT for training, which in principle can be developed for any number of variables. It remains yet to be studied, how adding more variables to the LUT parameter space affects the accuracy of the emulator. This will also depend on the role these variables play in driving the spectral output [32].

Altogether, considering all strengths and weaknesses of both interpolation and emulation, this study leads us concluding that the emulation technique can become an attractive alternative to interpolation in sampling an LUT parameter space. Based on the results presented here, it is foreseen that relying on emulation rather than interpolation will lead to more accurate, and typically faster querying through an LUT parameter space. This bears consequences in various RTM-based processing applications in which high accuracy and fast run-time is needed, e.g., in scene generation [32], [97], [98], in atmospheric correction procedures, in LUT-based inversion of biophysical parameters [99], [100], in hyperspectral target detection [101] or in instrument performance modeling [102]. Further studies are required to consolidate whether emulation techniques are always to be trusted as functioning more accurately than interpolation techniques.

VI. CONCLUSION

Computationally expensive RTMs are commonly used in various remote sensing applications. Because these RTMs take long processing time, the common practice is develop one time an LUT and then making use of interpolation techniques to sample the LUT parameter space. However, the question arose whether these interpolation techniques are most accurate. This paper proposed to use emulation, i.e., statistical learning, as an alternative to interpolation. Two experiments were conducted to ascertain

the accuracy in delivering spectral outputs of both techniques: one for TOC reflectance as generated by PROSAIL, and another for TOA radiance data as generated by MODTRAN. The interpolation and emulation methods were evaluated against a reference LUT of 5000 simulations, leading to the following results.

- 1) In all experiments the emulation methods clearly produced output spectra more accurately than the tested interpolation techniques.
- 2) GPR reproduced RTM output spectra up to ten times more accurately than interpolation methods and this with a speed that is <5% of the linear interpolation method, i.e., in mere seconds. The KRR was the second most accurate method, and this emulator is extremely fast: 5000 spectra were produced in a fraction of a second. Hence, the KRR shows an attractive tradeoff between accuracy and computational time.
- 3) Regarding emulation, some little more gain can be achieved by training with more samples or adding more PCA components in the regression model. However, for GPR this is at the expense of somewhat slowing down processing. It is thus concluded that emulation methods offer a better alternative in computational cost and accuracy than traditional methods based on interpolation to sample an LUT parameter space.

Future work will aim to include GPR emulators as an alternative to the current LUT interpolation methods implemented in the FLEX end-to-end mission simulator [9], [98]. This will likely reduce the computation time for the generation of synthetic scenes, which will extend the current FLEX simulator capabilities to perform sensitivity analysis for various leaf/canopy and atmospheric conditions. In addition, two research lines are currently being explored to improve the accuracy of interpolation/emulation methods. On the one hand, we are optimizing the LUT nodes distribution in order to reduce LUT size while increasing the accuracy of interpolation methods [103], [104]. On the other hand, we can improve the accuracy of statistical methods to generate multioutput emulators by using advanced machine learning methods, deep learning algorithms, or combining emulators, e.g., for different wavelength regions [105]. Both research lines will turn into more efficient sampling methods, reducing both computation burden and RTM-output reconstruction error.

REFERENCES

- [1] C. Mobley, *Light and Water: Radiative Transfer in Natural Waters*. New York, NY, USA: Academic, 1994, [Online] <http://www.sequoiasci.com/product/hydrolight>
- [2] Y. Zhang, W. Rossow, A. Lacis, V. Oinas, and M. Mishchenko, "Calculation of radiative fluxes from the surface to top of atmosphere based on ISCCP and other global data sets: Refinements of the radiative transfer model and the input data," *J. Geophys. Res. D, Atmos.*, vol. 109, no. 19, pp. 1–27, 2004.
- [3] S. Jacquemoud *et al.*, "PROSPECT + SAIL models: A review of use for vegetation characterization," *Remote Sens. Environ.*, vol. 113, pp. S56–S66, 2009.
- [4] M. Deiveegan, C. Balaji, and S. Venkateshan, "A polarized microwave radiative transfer model for passive remote sensing," *Atmos. Res.*, vol. 88, no. 3/4, pp. 277–293, 2008.
- [5] C. Van der Tol, J. A. Berry, P. K. E. Campbell, and U. Rascher, "Models of fluorescence and photosynthesis for interpreting measurements of solar-induced chlorophyll fluorescence," *J. Geophys. Res., Biogeosci.*, vol. 119, no. 12, pp. 2312–2327, 2014.
- [6] J. Verrelst *et al.*, "Optical remote sensing and the retrieval of terrestrial vegetation bio-geophysical properties—A review," *ISPRS J. Photogrammetry Remote Sens.*, vol. 108, pp. 273–290, 2015.
- [7] W. Verhoef and H. Bach, "Simulation of Sentinel-3 images by four-stream surface-atmosphere radiative transfer modeling in the optical and thermal domains," *Remote Sens. Environ.*, vol. 120, pp. 197–207, 2012.
- [8] K. Meharrar and N. Bachari, "Modelling of radiative transfer of natural surfaces in the solar radiation spectrum: Development of a satellite data simulator (SDDS)," *Int. J. Remote Sens.*, vol. 35, no. 4, pp. 1199–1216, 2014.
- [9] J. Vicent *et al.*, "FLEX end-to-end mission performance simulator," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 4215–4223, Jul. 2016.
- [10] J. Verrelst, J. Rivera, C. Van Der Tol, F. Magnani, G. Mohammed, and J. Moreno, "Global sensitivity analysis of the scope model: What drives simulated canopy-leaving sun-induced fluorescence?," *Remote Sens. Environ.*, vol. 166, pp. 8–21, 2015.
- [11] J. B. Feret *et al.*, "PROSPECT-4 and 5: Advances in the leaf optical properties model separating photosynthetic pigments," *Remote Sens. Environ.*, vol. 112, no. 6, pp. 3030–3043, 2008.
- [12] W. Verhoef, "Light scattering by leaf layers with application to canopy reflectance modeling: The SAIL model," *Remote Sens. Environ.*, vol. 16, no. 2, pp. 125–141, 1984.
- [13] H. Rahman and G. Dedieu, "SMAC: A simplified method for the atmospheric correction of satellite measurements in the solar spectrum," *Int. J. Remote Sens.*, vol. 15, no. 1, pp. 123–143, 1994.
- [14] A. Lipton, J.-L. Moncet, S.-A. Boukabara, G. Uymin, and K. Quinn, "Fast and accurate radiative transfer in the microwave with optimum spectral sampling," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 7, pp. 1909–1917, Jul. 2009.
- [15] Y. M. Govaerts and M. M. Verstraete, "Raytran: A Monte Carlo ray-tracing model to compute light scattering in three-dimensional heterogeneous media," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 2, pp. 493–505, Mar. 1998.
- [16] P. North, "Three-dimensional forest light interaction model using a Monte Carlo method," *IEEE Trans. Geosci. Remote Sens.*, vol. 34, no. 4, pp. 946–956, Jul. 1996.
- [17] P. Lewis, "Three-dimensional plant modelling for remote sensing simulation studies using the botanical plant modelling system," *Agronomie*, vol. 19, no. 3/4, pp. 185–210, 1999.
- [18] J. Gastellu-Etchegorry, V. Demarez, V. Pinel, and F. Zagolski, "Modeling radiative transfer in heterogeneous 3-D vegetation canopies," *Remote Sens. Environ.*, vol. 58, no. 2, pp. 131–156, 1996.
- [19] G. Petropoulos, M. Wooster, T. Carlson, M. Kennedy, and M. Scholze, "A global Bayesian sensitivity analysis of the 1D SimSphere soil vegetation atmospheric transfer (SVAT) model using Gaussian model emulation," *Ecol. Model.*, vol. 220, no. 19, pp. 2427–2440, 2009.
- [20] C. Van Der Tol, W. Verhoef, J. Timmermans, A. Verhoef, and Z. Su, "An integrated model of soil-canopy spectral radiances, photosynthesis, fluorescence, temperature and energy balance," *Biogeosciences*, vol. 6, no. 12, pp. 3109–3129, 2009.
- [21] A. Berk *et al.*, "MODTRANTM5: 2006 update," *Proc. SPIE*, vol. 6233, 2006, Art. no. 62331F.
- [22] M. España, F. Baret, F. Aries, M. Chelle, B. Andrieu, and L. Prévot, "Modeling maize canopy 3d architecture: Application to reflectance simulation," *Ecol. Model.*, vol. 122, no. 1/2, pp. 25–43, 1999.
- [23] M. Abramowitz and I. Stegun, "Handbook of mathematical functions," in *Applied Mathematics Series.*, vol. 55, Washington, DC, USA: Nat. Bureau Std., 1964, ch. 25.2.
- [24] I. Amidror, "Scattered data interpolation methods for electronic imaging systems: A survey," *J. Electron. Imag.*, vol. 11, no. 2, pp. 168–176, 2002.
- [25] E. Ientilucci and P. Bajorski, "Stochastic modeling of physically derived signature spaces," *J. Appl. Remote Sens.*, vol. 2, no. 1, p. 023532, 2008.
- [26] L. Guanter, R. Richter, and H. Kaufmann, "On the application of the MODTRAN4 atmospheric radiative transfer code to optical remote sensing," *Int. J. Remote Sens.*, vol. 30, no. 6, pp. 1407–1424, 2009.
- [27] A. Lyapustin, J. Martonchik, Y. Wang, I. Laszlo, and S. Korkin, "Multi-angle implementation of atmospheric correction (MAIAC): 1. Radiative transfer basis and look-up tables," *J. Geophys. Res. Atmospheres*, vol. 116, no. 3, p. D03210, 2011.

- [28] L. Scheck, P. Frèrebeau, R. Buras-Schnell, and B. Mayer, "A fast radiative transfer method for the simulation of visible satellite imagery," *J. Quant. Spectrosc. Radiat. Transf.*, vol. 175, pp. 54–67, 2016.
- [29] A. O'Hagan, "Bayesian analysis of computer code outputs: A tutorial," *Rel. Eng. Syst. Saf.*, vol. 91, no. 10–11, pp. 1290–1300, 2006.
- [30] J. L. Gómez-Dans, P. E. Lewis, and M. Disney, "Efficient emulation of radiative transfer codes using Gaussian processes and application to land surface parameter inferences," *Remote Sens.*, vol. 8, no. 2, p. 119, 2016.
- [31] J. Verrelst *et al.*, "Emulation of leaf, canopy and atmosphere radiative transfer models for fast global sensitivity analysis," *Remote Sens.*, vol. 8, no. 8, p. 673, 2016.
- [32] J. Verrelst, J. Rivera Caicedo, J. Muñoz Marí, G. Camps-Valls, and J. Moreno, "SCOPE-based emulators for fast generation of synthetic canopy reflectance and sun-induced fluorescence spectra," *Remote Sens.*, vol. 9, no. 9, p. 927, 2017.
- [33] D. Busby, "Hierarchical adaptive experimental design for Gaussian process emulators," *Rel. Eng. Syst. Safety*, vol. 94, no. 7, pp. 1183–1193, 2009.
- [34] Y.-J. Kim, "Comparative study of surrogate models for uncertainty quantification of building energy model: Gaussian process emulator vs. polynomial chaos expansion," *Energy Buildings*, vol. 133, pp. 46–58, 2016.
- [35] S. Razavi, B. A. Tolson, and D. H. Burn, "Numerical assessment of meta-modelling strategies in computationally intensive optimization," *Environ. Model. Softw.*, vol. 34, pp. 67–86, 2012.
- [36] L. S. Bastos and A. O'Hagan, "Diagnostics for Gaussian process emulators," *Technometrics*, vol. 51, no. 4, pp. 425–438, 2009.
- [37] A. O'Hagan, "Probabilistic uncertainty specification: Overview, elaboration techniques and their application to a mechanistic model of carbon flux," *Environ. Model. Softw.*, vol. 36, pp. 35–48, 2012.
- [38] N. E. Owen, P. Challenor, P. P. Menon, and S. Bennani, "Comparison of surrogate-based uncertainty quantification methods for computationally expensive simulators," *SIAM/ASA J. Uncertainty Quantification*, vol. 5, pp. 403–435, 2015.
- [39] J. Gastellu-Etchegorry, F. Gascon, and P. Esteve, "An interpolation procedure for generalizing a look-up table inversion method," *Remote Sens. Environ.*, vol. 87, no. 1, pp. 55–71, 2003.
- [40] D. Shepard, "Two-dimensional interpolation function for irregularly-spaced data," in *Proc. 23rd Nat. Conf.*, 1968, pp. 517–524.
- [41] S. Łukaszuk, "A new concept of probability metric and its applications in approximation of scattered data sets," *Comput. Mech.*, vol. 33, no. 4, pp. 299–304, Mar. 2004.
- [42] T. Cooley *et al.*, "FLAASH, a MODTRAN4-based atmospheric correction algorithm, its applications and validation," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2002, vol. 3, pp. 1414–1418.
- [43] R. Richter and D. Schlöpfer, "Geo-atmospheric processing of airborne imaging spectrometry data. Part 2: Atmospheric/topographic correction," *Int. J. Remote Sens.*, vol. 23, no. 13, pp. 2631–2649, 2002.
- [44] C. Barber, D. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.
- [45] B. Delaunay, "Sur la sphère vide. A la mémoire de Georges Voronoï," *Bull. de l'Académie Des Sci. de l'URSS. Classe Des Sci. Mathématiques et na.*, no. 6, pp. 793–800, 1934.
- [46] R. Sibson, "A brief description of natural neighbor interpolation," in *Interpolating Multivariate Data*. New York, NY, USA: Wiley, 1989, ch. 2, pp. 21–36.
- [47] I. The MathWorks, "Interpolate N-D scattered data," 2017, R2017a. [Online]. Available: <https://es.mathworks.com/help/matlab/ref/griddatan.html>
- [48] S. W. Park, L. Linsen, O. Kreylos, J. D. Owens, and B. Hamann, "Discrete Sibson interpolation," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 2, pp. 243–253, Mar./Apr. 2006.
- [49] B. J. Bartels, R. H. and B. Barsky, "Hermite and cubic spline interpolation," in *An Introduction Splines Use Computer Graphics and Geometric Modelling*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann., 1998, ch. 3, pp. 9–17.
- [50] J. Rohmer and E. Foerster, "Global sensitivity analysis of large-scale numerical landslide models based on Gaussian-process meta-modeling," *Comput. Geosci.*, vol. 37, no. 7, pp. 917–927, 2011.
- [51] C. Carnevale, G. Finzi, G. Guariso, E. Pisoni, and M. Volta, "Surrogate models to compute optimal air quality planning policies at a regional scale," *Environ. Model. Softw.*, vol. 34, pp. 44–50, 2012.
- [52] N. Villa-Vialaneix, M. Follador, M. Ratto, and A. Leip, "A comparison of eight metamodelling techniques for the simulation of N₂ O fluxes and n leaching from corn crops," *Environ. Model. Softw.*, vol. 34, pp. 51–66, 2012.
- [53] A. Castelletti, S. Galelli, M. Ratto, R. Soncini-Sessa, and P. Young, "A general framework for dynamic emulation modelling in environmental problems," *Environ. Model. Softw.*, vol. 34, pp. 5–18, 2012.
- [54] L. Lee *et al.*, "The magnitude and causes of uncertainty in global model simulations of cloud condensation nuclei," *Atmos. Chem. Phys.*, vol. 13, no. 17, pp. 8879–8914, 2013.
- [55] N. Bounceur, M. Crucifix, and R. Wilkinson, "Global sensitivity analysis of the climate–vegetation system to astronomical forcing: an emulator-based approach," *Earth Syst. Dyn. Discuss.*, vol. 5, no. 2, pp. 901–943, 2014.
- [56] G. Ireland, G. Petropoulos, T. Carlson, and S. Purdy, "Addressing the ability of a land biosphere model to predict key biophysical vegetation characterisation parameters with global sensitivity analysis," *Environ. Model. Softw.*, vol. 65, pp. 94–107, 2015.
- [57] M. McKay, R. Beckman, and W. Conover, "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.
- [58] F. Baret, J. Clevers, and M. Steven, "The robustness of canopy gap fraction estimates from red and near-infrared reflectances: A comparison of approaches," *Remote Sens. Environ.*, vol. 54, no. 2, pp. 141–151, 1995.
- [59] B. Combal *et al.*, "Retrieval of canopy biophysical variables from bidirectional reflectance using prior information to solve the ill-posed inverse problem," *Remote Sens. Environ.*, vol. 84, no. 1, pp. 1–15, 2003.
- [60] J. Verrelst *et al.*, "Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for Sentinel-2 and -3," *Remote Sens. Environ.*, vol. 118, pp. 127–139, 2012.
- [61] J. Rivera, J. Verrelst, J. Delegido, F. Veroustraete, and J. Moreno, "On the semi-automatic retrieval of biophysical parameters based on spectral index optimization," *Remote Sens.*, vol. 6, no. 6, pp. 4924–4951, 2014.
- [62] R. K. Hankin, "Introducing BACCO, an R package for Bayesian analysis of computer code output," *J. Stat. Softw.*, vol. 14, no. 16, pp. 1–21, 2005.
- [63] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, Jan. 1968.
- [64] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, no. 1–3, pp. 37–52, 1987.
- [65] J. P. Rivera, J. Verrelst, J. Gómez-Dans, J. Muñoz Marí, J. Moreno, and G. Camps-Valls, "An emulator toolbox to approximate radiative transfer models with statistical learning," *Remote Sens.*, vol. 7, no. 7, p. 9347, 2015.
- [66] S. Conti, J. Gosling, J. Oakley, and A. O'Hagan, "Gaussian process emulation of dynamic computer codes," *Biometrika*, vol. 96, no. 3, pp. 663–676, 2009.
- [67] F. Liu and M. West, "A dynamic modelling strategy for Bayesian computer model emulation," *Bayesian Anal.*, vol. 4, no. 2, pp. 393–412, 2009.
- [68] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.
- [69] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [70] G. Camps-Valls and L. Bruzzone, Eds., *Kernel Methods for Remote Sensing Data Analysis*. Chichester, U.K.: Wiley, Dec. 2009.
- [71] J. Rojo-Álvarez, M. Martínez-Ramón, J. Muñoz Marí, and G. Camps-Valls, *Digital Signal Processing With Kernel Methods*. Chichester, U.K.: Wiley, Apr. 2017.
- [72] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. New York, NY, USA: MIT Press, 2006.
- [73] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
- [74] S. Haykin, *Neural Networks—Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, Oct. 1999.
- [75] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford Univ. Press, 1995.
- [76] G. S. Kimeldorf and G. Wahba, "A correspondence between Bayesian estimation on stochastic processes and smoothing by splines," *Ann. Math. Statist.*, vol. 41, no. 2, pp. 495–502, 1970.
- [77] J. Verrelst, E. Romijn, and L. Kooistra, "Mapping vegetation density in a heterogeneous river floodplain ecosystem using pointable CHRIS/PROBA data," *Remote Sens.*, vol. 4, no. 9, pp. 2866–2889, 2012.

- [78] J. Vicent, N. Sabater, J. Verrelst, L. Alonso, and J. Moreno, "Assessment of approximations in aerosol optical properties and vertical distribution into FLEX atmospherically-corrected surface reflectance and retrieved sun-induced fluorescence," *Remote Sens.*, vol. 9, no. 7, p. 675, 2017.
- [79] S. Jacquemoud *et al.*, "PROSPECT + SAIL models: A review of use for vegetation characterization," *Remote Sens. Environ.*, vol. 113, pp. S56–S66, 2009.
- [80] W. Verhoef, L. Jia, Q. Xiao, and Z. Su, "Unified optical-thermal four-stream radiative transfer theory for homogeneous vegetation canopies," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1808–1822, Jun. 2007.
- [81] K. Stamnes, S.-C. Tsay, W. Wiscombe, and K. Jayaweera, "Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media," *Appl. Opt.*, vol. 27, no. 12, pp. 2502–2509, Jun. 1988.
- [82] R. Goody, R. West, L. Chen, and D. Crisp, "The correlated-k method for radiation calculations in nonhomogeneous atmospheres," *J. Quant. Spectrosc. Radiat. Transf.*, vol. 42, no. 6, pp. 539–550, 1989.
- [83] A. Baldridge, S. Hook, C. Grove, and G. Rivera, "The ASTER spectral library version 2.0," *Remote Sens. Environ.*, vol. 113, no. 4, pp. 711–715, 2009.
- [84] R. Richter, "A spatially adaptive fast atmospheric correction algorithm," *Int. J. Remote Sens.*, vol. 17, no. 6, pp. 1201–1214, 1996.
- [85] A. O'Hagan, *Bayesian Inference* (Kendall's Advanced Theory of Statistics, vol. 2B. London, U.K.: Arnold, 1994.
- [86] J. Oakley and A. O'Hagan, "Probabilistic sensitivity analysis of complex models: A Bayesian approach," *J. Roy. Statist. Soc. Series B, Statist. Methodol.*, vol. 66, no. 3, pp. 751–769, 2004.
- [87] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.
- [88] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.
- [89] M. U. Gutmann and J. Corander, "Bayesian optimization for likelihood-free inference of simulator-based statistical models," *J. Mach. Learn. Res.*, vol. 16, pp. 4256–4302, 2015.
- [90] J. Mockus, *Bayesian Approach to Global Optimization*. Dordrecht, The Netherlands: Kluwer, 1989.
- [91] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 2951–2959.
- [92] S. K. Kirkpatrick, C. D. Gelatt, Jr, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
- [93] L. Martino, V. Elvira, D. Luengo, J. Corander, and F. Louzada, "Orthogonal parallel MCMC methods for sampling and optimization," *Digital Signal Process.*, vol. 58, pp. 64–84, 2016.
- [94] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York, NY, USA: Springer, 2004.
- [95] L. Martino and V. Elvira, *Metropolis Sampling*. New York, NY, USA: Wiley, 2017.
- [96] L. Martino and J. Read, "A multi-point Metropolis scheme with generic weight functions," *Statist. Probability Lett.*, vol. 82, no. 7, pp. 1445–1453, 2012.
- [97] E. Ientilucci and S. Brown, "Advances in wide area hyperspectral image simulation," *Proc. SPIE*, vol. 5075, 2003, pp. 110–121.
- [98] C. Tenjo *et al.*, "Design of a generic 3D Scene Generator for passive optical missions and its implementation for the ESA's FLEX/Sentinel-3 tandem mission," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 13, pp. 1290–1307, Mar. 2017.
- [99] J. Rivera, J. Verrelst, G. Leonenko, and J. Moreno, "Multiple cost functions and regularization options for improved retrieval of leaf chlorophyll content and LAI through inversion of the PROSAIL model," *Remote Sens.*, vol. 5, no. 7, pp. 3280–3304, 2013.
- [100] J. Verrelst, J. Rivera, G. Leonenko, L. Alonso, and J. Moreno, "Optimizing LUT-Based RTM inversion for semiautomatic mapping of crop biophysical parameters from sentinel-2 and -3 Data: Role of cost functions," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 257–269, Jan. 2014.
- [101] S. Matteoli, E. Ientilucci, and J. Kerekes, "Operational and performance considerations of radiative-transfer modeling in hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 4, pp. 1343–1355, Apr. 2011.
- [102] J. Kerekes and J. Baum, "Full-spectrum spectral imaging system analytical model," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 571–580, Mar. 2005.
- [103] L. Martino, J. Vicent, and G. Camps-Valls, "Automatic emulator and optimized look-up table generation for radiative transfer models," in *Proc. Int. Geosci. Remote Sens. Symp.*, Jul. 2017, pp. 1457–1460.
- [104] J. Vicent *et al.*, "Gradient-based automatic look-up table generator for radiative transfer models," *IEEE Trans. Geosci. Remote Sens.*, vol. (preprint), pp. 1–9, 2018, doi: [10.1109/TGRS.2018.2864517](https://doi.org/10.1109/TGRS.2018.2864517).
- [105] D. Heestermans-Svendsen, L. Martino, J. Vicent, and G. Camps-Valls, "Multioutput automatic emulator for radiative transfer models," in *Proc. Int. Geosci. Remote Sens. Symp.*, Jul. 2018.



Jorge Vicent received the B.Sc. degree in physics from the University of Valencia, Valencia, Spain, in 2008, the M.Sc. degree in physics from the École polytechnique fédérale de Lausanne, Lausanne, Switzerland, in 2010, and the Ph.D. degree in remote sensing from the University of Valencia, Valencia, Spain, in 2016.

Since November 2017, he has been with Magellium with the Department of Earth Observation, Toulouse, France as a R&D Engineer. He is currently involved in developing the Level-2 processing chain for ESA's FLEX mission. His research interests include the modeling of earth observation satellites, system engineering, radiative transfer modelling, atmospheric correction, and hyperspectral data analysis.



Jochem Verrelst received the M.Sc. degree in tropical land use and in geo-information science both in 2005 and the Ph.D. degree in remote sensing in 2010 from Wageningen University, Wageningen, Netherlands. His dissertation focused on the space-borne spectrodirectional estimation of forest properties.

Since 2010, he has been involved in preparatory activities of FLEX. He is the founder of the ARTMO software package. In 2017, he received a H2020 ERC Starting Grant (#755617) to work on the development of vegetation products based on synergy of FLEX and

Sentinel-3 data. His research interests include retrieval of vegetation properties using airborne and satellite data, canopy radiative transfer modeling and emulation, and hyperspectral data analysis.



Juan Pablo Rivera-Caicedo received the B.Sc. degree in agricultural engineering from the University National of Colombia and University of Valle, Cali, Colombia, in 2001, the Master's degree in irrigation engineering from the CEDEX-Centro de Estudios y Experimentación de Obra Públicas, Madrid, Spain, in 2003, and the M.Sc. and Ph.D. degrees in remote sensing from the University of Valencia, Valencia, Spain, in 2011 and 2014, respectively.

Since January 2011, he has been a member of the Laboratory for Earth Observation, Image Processing Laboratory, University of Valencia, Spain. Since 2016, he has been with the Concejo Nacional de Ciencia y Tecnología—CONACYT in México in the program: Catedras-Conacyt. He is currently involved in preparatory activities of the Fluorescence Explorer. His research interests include retrieval of vegetation properties using airborne and satellite data, leaf and canopy radiative transfer modeling, and hyperspectral data analysis.



Neus Sabater received the B.Sc. degree in physics in 2010, and the M.Sc. and Ph.D. degrees in remote sensing in 2018 from the Universitat de València, Valencia, Spain.

Since August 2012, she has been involved in the activities of the Laboratory for Earth Observation at the Image Processing Laboratory, University of Valencia, as a Research Technician. Main activities during this period were related to the development of the preparatory activities of the FLEX mission. Her research interests include atmospheric correction, atmospheric radiative transfer, meteorology, and hyperspectral RS.

Dr. Sabater was the recipient of awarded a Ph.D. scholarship from the Spanish Ministry of Economy and Competitiveness, associated to the Ingenio/Seosat Spanish space mission in 2013 and also was the recipient of the University of Valencia to the best student records in the M.Sc. of remote sensing (2012–2013).



Jordi Muñoz-Mari (M'11) received the B.Sc. degree in physics in 1993, and the B.Sc. and Ph.D. degrees in electronics engineering in 1996 and 2003, respectively, from the Universitat de València, Valencia, Spain.

He is currently an Associate Professor with the Department of Electronics Engineering, Universitat de València, where he teaches electronic circuits and, programmable logical devices, digital electronic systems and microprocessor electronic systems. His research interests include development

of machine learning algorithms for signal and image processing. Please visit <http://www.uv.es/jordi/> for more information.



Gustau Camps-Valls (M'04–SM'07–F'18) received the Ph.D. degree in physics from the Universitat de València, Valencia, Spain, in 2002.

He is currently a Full Professor in electrical engineering, and a Coordinator of the Image and Signal Processing Group, Universitat de València. He is interested in the development of machine learning algorithms for geoscience and remote sensing data analysis. He entered the list of highly cited researchers by Thomson Reuters in 2011, holds an $h = 55$, has authored and coauthored more than 150 journal papers,

200 conference papers, and 4 books on machine learning, remote sensing and signal processing.

Dr. Camps-Valls was the recipient of the prestigious ERC Consolidator Grant to advance in statistical inference for Earth observation data analysis in 2015. He is an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING LETTERS, and IEEE GEOSCIENCE AND REMOTE SENSING LETTERS. Visit <http://isp.uv.es> for more information.



José Moreno (M'18) is currently Ph.D degree in theoretical physics with the Department of Earth Physics and Thermodynamics, Faculty of Physics, University of Valencia, Valencia, Spain, as a Professor of earth physics, teaching and working on different projects related to remote sensing and space research as responsible for the Laboratory for Earth Observation. His main work is related to the modeling and monitoring of land surface processes by using remote sensing techniques. He has been involved in many international projects and research networks, including the preparatory activities and exploitation programmes of several satellite missions (ENVISAT, CHRIS/PROBA, GMES/Sentinels, SEOSAT) and the Fluorescence Explorer (FLEX), ESA's 8th Earth Explorer mission.

Dr. Moreno was an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (1994–2000) and has been a member of the ESA Earth Sciences Advisory Committee (1998–2002), the Space Station Users Panel, and other international advisory committees. He is the Director of the Laboratory for Earth Observation with the Image Processing Laboratory/Scientific Park.

Dr. Moreno was an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (1994–2000) and has been a member of the ESA Earth Sciences Advisory Committee (1998–2002), the Space Station Users Panel, and other international advisory committees. He is the Director of the Laboratory for Earth Observation with the Image Processing Laboratory/Scientific Park.