esa	**HR-AlbedoMap Algorithm Theoretical Basis Document** **Author:** J-P Muller	UCL

TITLE: HR-AlbedoMap Algorithm Theoretical Basis Document

Document Number: HR-AlbedoMap_ATBD-v1, Date 25/06/2021

Distribution:

ESA	E. Volden	X
	F. Ramoino	X
University College London	J-P. Muller	X
	R. Song	X
	A. Francis	X

Author: Prof. J-P Muller ________________ Date: 25/06/2021 ________________

Manager/Project Office Jan-Peter Muller ________________ Date: 25/06/2021 ________________

PA: ________________ Date: ________________

CHANGE RECORD

ISSUE	DATE	PAGES CHANGED	COMMENTS
1	31/3/21	All new	First Issue
1.1	25/06/21	30-37	Second issue to include verification report

CONTRIBUTORS

Author Names	Organisation
J.-P. Muller, R. Song, A. Francis	UCL MSSL

CONTENTS

1 Introduction

1.1 Scope of the Document

The ESA Science for Society **High-Resolution AlbedoMap** (known hereafter as HR-Albedo) project aims to develop a surface albedo generation system that can produce high temporal and high resolution spatial surface spectral and broadband albedo products, which offer great potential for mapping land surface energy budgets especially over agricultural and forest areas. In addition, this high-resolution albedo product can improve the performance of global climate models by providing better constraints on surface albedo parameterisation.

This Algorithm Theoretical Basis Document (ATBD) introduces the following 5 modules inside this high-resolution albedo generation framework, which include the 1) machine learning based cloud detection module; 2) Sensor invariant Atmospheric Correction (SIAC) module; 3) Endmember-based Sentinel-2 pure pixel extraction module; 4) high-resolution albedo retrieval module; and 5) albedo gap filling module.

1.2 Acronyms

The definition of the acronyms used in this document is provided hereafter:

ATBD	Algorithm Theoretical Basis Document
AOT	Aerosol Optical Thickness
ATGP	Automatic Target Generation Process
BHR	Bi-Hemispherical diffuse Reflectance
BoA	Bottom-of-Atmosphere
BRDF	Bidirectional Reflectance Distribution Function
BRF	Bidirectional Reflectance Factor
BSA	Black Sky Albedo
CAMS	Copernicus Atmospheric Monitoring Service
DHR	Directional Hemispherical Reflectance
FCLS	Fully Constrained Least Squares
RTLSR	RossThick-LiSparse-Reciprocal
SIAC	Sensor invariant Atmospheric Correction
PDF	Probability Density Function
PSF	Point Spread Function
ToA	Top-of-Atmosphere
TCO3	Total Columnar Ozone
TCWV	Total Columnar Water Vapour
WSA	White Sky Albedo

2 High-Resolution Albedo Generation System

2.1 Overview

The High-Resolution AlbedoMap (HR-Albedo) framework aims to provide 10-m albedo products for Sentinel-2 band-2 (492.4 nm), band-3 (559.8 nm), band-4 (664.6 nm), broadband VIS [400nm, 700nm], and 20-m albedo products for band-8A (864.7 nm), band-11 (1613 nm), band 12 (2202 nm), broadband NIR [700nm, 4000nm] and broadband SW [300nm, 4000nm]. The albedo products consist of two parts, which are Directional Hemispherical Reflectance (DHR), also known as Black-Sky-Albedo (BSA), and Bi-Hemispherical diffuse Reflectance (BHR), also known as White-Sky-Albedo (WSA).

Figure 1 shows an overview of the processing chain for the HR-Albedo framework.

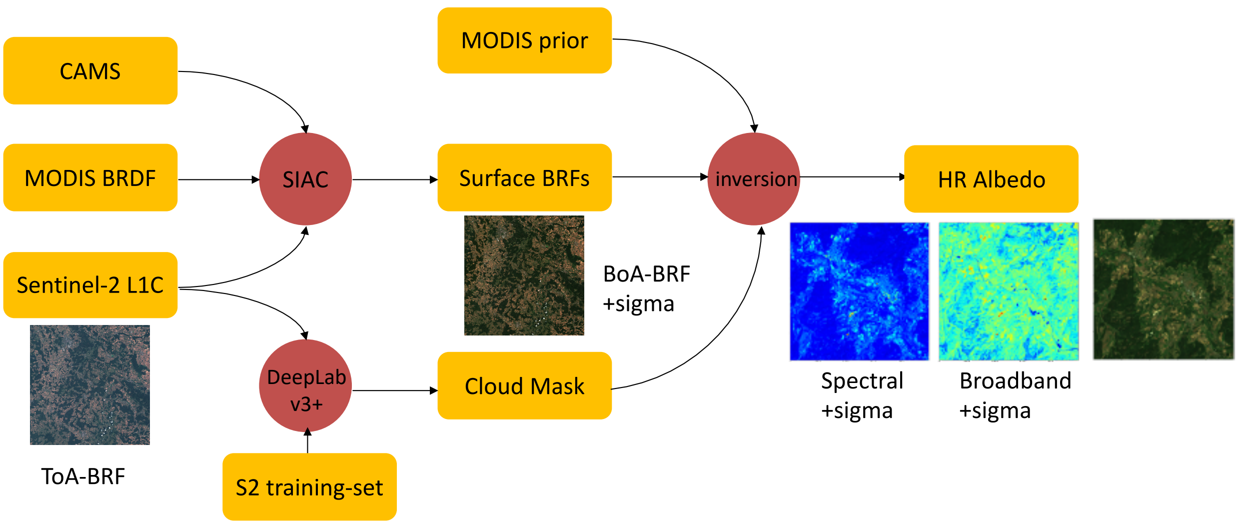


Figure 1. Overall Processing chain of the HR-AlbedoMap system.

Input data to the level-1C ToA-BRF is fed in parallel to the cloud masking system (DeepLabV3) and the Atmospheric Correction module (SIAC) [33] which itself uses the daily MODIS BRDF and the background aerosol field (CAMS) [34] to produce surface BRFs which over vegetated surfaces are called BoA. The separately calculated cloud mask and BoA are then employed so that the BRF retrieval is only performed over cloud-free areas and inverted using an Endmember extraction employing all spectral channels of interest to produce the HR-albedo product. Not shown here is what happens if there is cloud/cloud-shadow obscuration which is described below.

More details on the individual elements within this overall system are shown in Figure 2 and discussed further in successive sub-sections.

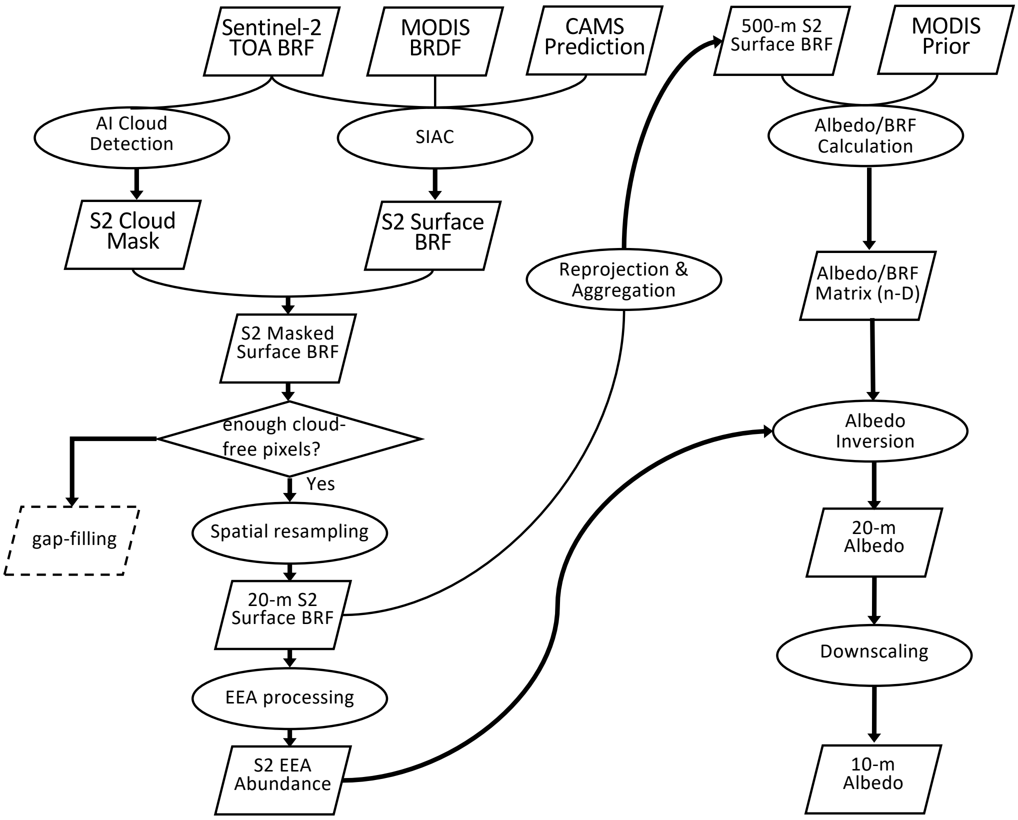


Figure 2. Further details of processing steps
N.B. (EEA: Endmember Extraction Algorithm)

This section provides an in-depth description of all the processing stages, which the cloud mask, endmember extraction and abundance estimation, high-resolution albedo inversion, gap filling.

2.2 Practical Considerations

The estimated computing time for different modules as well as the dataset sizes are listed in Table 1.

Table 1. Estimated computation time for different modules on a VM machine, and size of input/output datasets.

Modules	Computing Time
Computing cloud mask over scene **GPU** (including time of data resampling)	24.2s
Computing cloud mask over scene **CPU** (including time of data resampling)	14min
Atmospheric Correction (excluding time of downloading MCD43A)	30min

High-Resolution Albedo Retrieval (without use of parallel computing)	270 mins (4.5 hours)
Datasets	**Size of datasets per tile**
S2 L1C (110 x 110km - Input)	920 Mbytes
MCD43A1 (10º x 10º - Input)	180 Mbytes
VNP43MA1 (10º x 10º - Input)	180 Mbytes
High-Resolution S2 Albedo (Output)	450 Mbytes

2.3 Output Products

The output products include 1) 10-m albedo in Sentinel-2 band-2, band-3, band-4 and converted broadband VIS; 2) 20-m albedo in Sentinel-2 band-8A, band-11, band-12, and converted broadband NIR and SW. The output products are in GeoTIFF format, which includes the albedo values, cloud mask, and gap-filling flag.

2.4 Cloud Masking

2.4.1 Introduction of Cloud Masking

Albedo retrieval relies on having a high confidence that the area being retrieved is not obstructed by clouds or their cast shadows. Even thin clouds can drastically alter the estimated albedo measurements. The ESA HR-AlbedoMap project relies on collecting albedo measurements over the same areas many times, and so a highly accurate, reliable, and fast cloud detection algorithm will help immensely in reducing the number of cloudy areas incorrectly used, and maximising the number of cloud-free areas retrieved. To this end, a component of the project was to develop a state-of-the-art cloud masking algorithm for Sentinel-2.

Deep learning algorithms---specifically, convolutional neural networks---offer high performance across a huge range of computer vision tasks. Cloud masking falls under the category of image segmentation (the prediction of classes across the image at each pixel). In recent years, many model architectures have been proposed that are suited for image segmentation tasks. DeepLab v3+ is a well-known and highly performant example of such a convolutional neural network. We employ DeepLab v3+ here, trained on our own dataset of hand-labelled Sentinel-2 images, to detect both cloud and cloud shadow.

Clouds often cast shadows on the surface, which affects albedo too. Therefore, it is important to include cloud shadows as a detected class. In this application, we do not particularly care if a pixel is cloudy or shadowed as both will affect albedo, and so in the final mask we combine these two classes to simply determine whether a pixel is valid for albedo retrieval or not. However, clouds and cloud shadows are considered separately by the model itself. We present performance for both the cloud vs. non-cloud model (which uses all available training data) and the clear vs. cloud/cloud-shadow (which uses a subset of the data for which shadow annotations could be made).

This section of the ATBD proceeds with a look at prior work on cloud masking, and then an overview of the algorithm's development and design. Then, there is a section on the dataset used in training and validation of the algorithm. Next, details of how DeepLab v3+ was trained is given. Then, the pre-processing operations and implementation of a sliding window, so that the model can be used on extended areas, is covered. Finally, performance results on the test dataset, and some computational benchmarks, are shown, with comparison to other models where possible.

2.4.2 Related Work

A wide variety of modelling approaches have been put forward to detect clouds [1]. Many algorithms are based on reflectance or radiance thresholds, which are specified by experts and are usually fixed or based on a time-varying background (e.g. ISCCP, https://isccp.giss.nasa.gov/). For example, Landsat 7's ACCA [2]estimates the cloud cover over a scene by passing the band values through a chain of if_then/else gates which predict whether each pixel is a cloud. Similarly, [3] use a thresholding method on the high-resolution visible channel of Meteosat SEVIRI, and Sen2Cor, which serves to correct for atmospheric effects including clouds in Sentinel-2 Level 2 products [4]. Fmask [5], [6] is an algorithm originally designed for Landsat imagery, which uses situational thresholding based on whether the cloud is thought to be above water or land, and can assign confidence values to its predictions, allowing users to balance their needs for sensitivity and specificity of detections. Unfortunately, in studies comparing the relative performance of different methods, thresholding based approaches tend to fall short of techniques which embrace some kind of optimisation through labelled training data [1], [7].

Convolutional Neural Networks (CNNs) have come to dominate research in many Computer Vision tasks, perhaps triggered by the publication of AlexNet [8], [9]. For image segmentation, many architectures have been proposed. For example, U-Net [10] was developed for biomedical image segmentation, but variations of it have also been used with success in cloud masking [11], [12], [13], [14]. More bespoke architectures have also been developed, e.g. [15], [16], [17], [18], [19].

2.4.3 Algorithm Overview

DeepLab v3+ [20] is a convolutional model, building on versions v1 [21], v2 [22], and v3 [23]. For comprehensive descriptions on the specific architecture of DeepLab v3+, the reader is recommended to refer to these papers. We base our implementation on the code of Jenia Golbstein (github commit here) as it defines the DeepLab v3+ architecture as a Keras model with compatibility with TensorFlow 2, which was the desired framework to use in the project, due to our own familiarity with it, and its extensive functionality. The full model definition file can be found in Appendix A, as it is not included in the final cloud masking

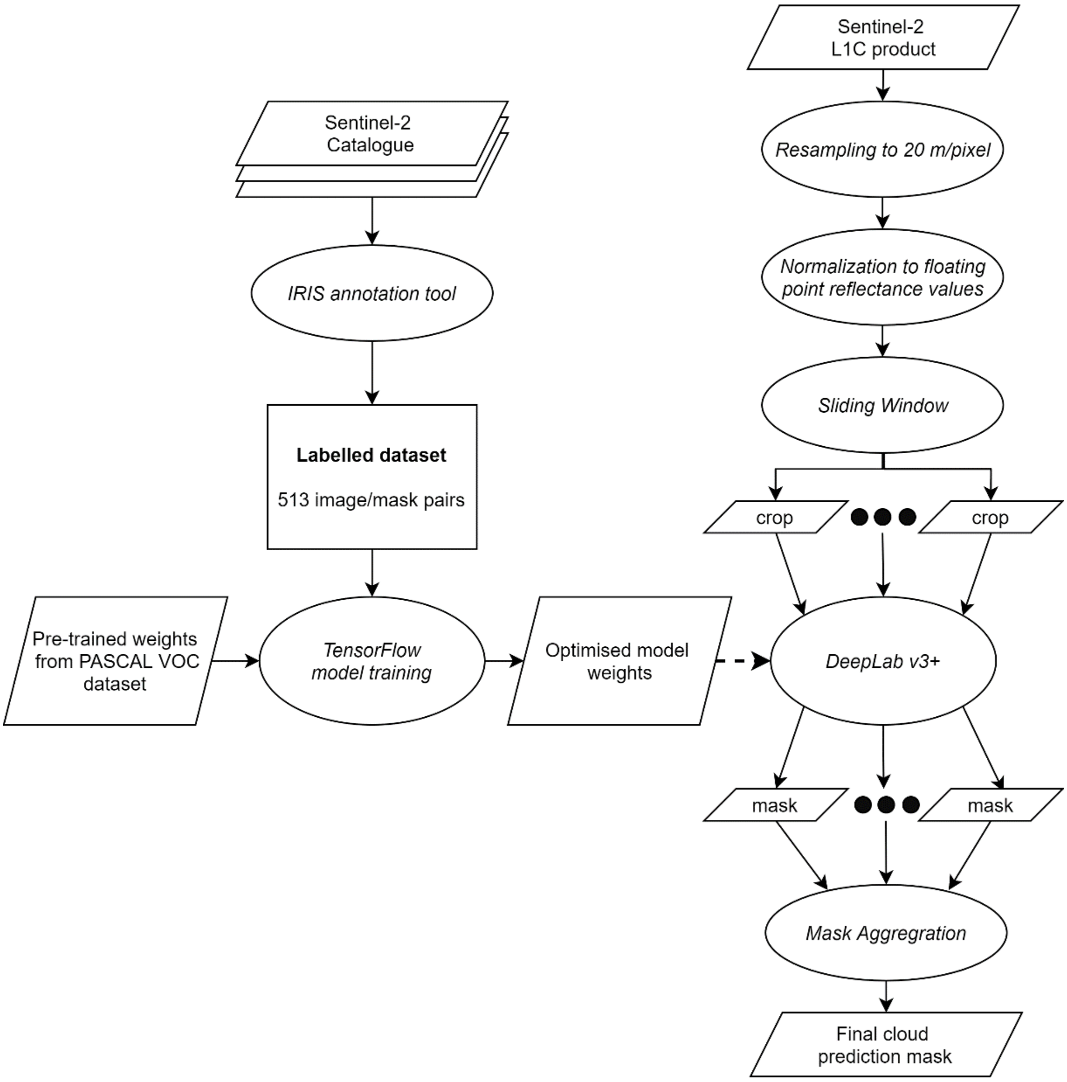


Figure 3. Flowchart of cloud masking algorithm. Column of operations to the left was done during the algorithm's development, whilst those on the right occur during algorithm operation

package (https://github.com/aliFrancis/Sen2DLv3).

DeepLabv3+ uses a large CNN (pre-trained on other tasks – in our case it was optimised during training on the PASCAL VOC dataset [24]) as a feature extractor, or “backbone”. Image segmentation performance often benefits from fusing information from different spatial scales, from the single pixel, to the whole image. This backbone is used to represent the spatial and spectral information in the image as high-level features, which it learns through training. In order for effective segmentation, features must represent a range of spatial scales. DeepLabv3+ achieves this through *atrous convolutions*. Atrous convolutions are like a standard convolution, but with a dilated kernel. The rate of dilation can then lead to features being extracted at different scales, by taking information from wider and wider fields of view. The resulting feature maps are combined using spatial pyramid pooling [25]. Whilst class predictions are required at every pixel, many neighbouring predictions are highly correlated with one another. Therefore, DeepLabv3+ outputs downsampled predictions, at a rate known as the *output stride*, and then upsamples the predictions to the full image resolution. The output stride is a power of two, and is often set to 8, 16, or 32.

In our work, we use an output stride of 8, given that cloud masking often requires very small areas to be segmented, unlike in other image segmentation domains where the output mask is less spatially complex. This relatively low output stride increases the computational requirements in training. For this reason, we use a relatively computationally cheap CNN as our feature extractor; MobileNetv2 [26], which enabled us to use a higher batch size in training than would be possible with larger CNNs.

Before input into DeepLab v3+, the data is pre-processed -- resampled to 20 m/pixel, normalised, and cut into windowed crops (Figure 3). After predictions by DeepLab v3+ are made, the individual masks are aggregated, leading to a final prediction with confidence values for each class at each pixel of the original Sentinel-2 L1C scene.

2.4.4 Sentinel-2 Dataset

The dataset used in training and testing is freely available here: doi:10.5281/zenodo.4172871. Full details of the annotation strategy, and dataset structure, can be found in the documentation there. Given that this dataset now exists, we do not believe it necessary to exhaustively describe its creation here, as it is available for anyone to use who wants to repeat or build on our work. However, here we reiterate the key points for context.

The dataset includes data from 513 L1C Sentinel-2 images, selected randomly from the entire 2018 Sentinel-2 archive (with each scene having an equal chance of inclusion). A 1022-by-1022 pixel area (at 20 m/pixel) was cropped from each product, with checks to make sure all pixels were valid (no no-data values were allowed).

Annotations were made using *IRIS* (github commit here), developed by Alistair Francis and John Mrziglod whilst at ESA’s Phi-Lab. This tool allowed us to rapidly inspect and annotate multispectral imagery, by displaying multiple band combinations side by side, and using a Random Forest [27] machine learning algorithm to extrapolate from one’s annotations on a subscene to fill the rest of that image with predicted labels/predictions. These predictions could then be iteratively corrected using an electronic eraser and/or adding further pixels of the same cloud or cloud shadow class and remade, allowing for a good balance between human accuracy and machine speed.

The annotations were made by both Alistair Francis and John Mrziglod, with roughly half the scenes done by each. At the beginning of the process, 10 scenes were identified as showing difficult or ambiguous features (e.g. thin Cirrus clouds). This was known as the "calibration set" and was marked by both annotators collaboratively, in order to approach a close agreement in annotation styles. At the end of annotation, we also selected 50 subscenes as a "validation set" which we both annotated separately, and subsequently measured the agreement between them (94.98% of pixels marked the same).

424 of the 513 scenes were marked with cloud shadow, if present, as well as cloud. However, the other 89 scenes displayed cloud shadow that was deemed too difficult to annotate, and so only includes cloud and clear classes. These difficulties stemmed from thin cloud shadows, which were very difficult to spot, and terrain shadow which got mixed with cloud shadow to make it impossible to distinguish between them. It was not possible to access the DEM employed for orthorectification to calculate which ones were terrain shadow (F. Gascon, private communication, 2021).

Each subscene was also placed into some non-mutually exclusive categorisations, which described the surface type, cloud type, cloud height, and cloud structure. These can later be used to measure performance against certain situations (e.g. Cumulus clouds over forested terrain). A full list of categories and their definitions is found in the dataset's documentation.

The dataset is distributed in numpy array format. The subscenes are floating point 1022-by-1022-by-13 dimension arrays, and the masks as boolean arrays of dimensions 1022-by-1022-by-3. The scene-wise categories are kept as a table in a .csv file.

2.4.5 Training of DeepLab v3+

DeepLab v3+ is trained using TensorFlow 2 and the Keras Model API (https://www.tensorflow.org/api_docs/python/tf/keras/Model). The training data used comes from the Sentinel-2 cloud mask dataset mentioned in the previous section. In the case of the cloud vs. non-cloud model, 206 of the 513 scenes are selected randomly to be included in training, and another 51 for validation. Similarly, for the 3-class (clear, cloud, cloud-shadow) model, 424 scenes are available which have those classes marked. 170 are used in training, 42 in validation, and the rest are kept for testing. This test set is identical to the 2-class one, but with those scenes without shadow markings removed.

During training, random images and masks are selected and read from disk. Then, data augmentation is applied in order to increase the variation of data, which increases model performance and attempts to prevent the model over-fitting. The augmentation routine consists of the steps described in Section 2.4.5.

Table 2: Data augmentation routine used during training. The effects are applied sequentially from the top down.

Effect	Probability of use	Notes
Cropping	100%	A 257-by-257 crop is randomly selected from across the image and masked
Rotation	100%	Rotates image by random multiples of 90^{o}
Flipping	50%	Reflects image along x-axis
Salt and pepper noise	50% chance	Randomly changes 1% of image pixel values to either 0 (pepper) or 1 (salt)
White noise	50% chance	Changes all values in the image by small normally distributed amount (standard deviation 0.05)
Band-wise multiplication	50% chance	Multiplies each image band by a different random value between 0.9-1.1

Before beginning the training, pre-trained weights are loaded into all model layers (except the first convolutional layer, as it has a different shape to the pre-trained model's, with 13 channels, instead of 3). These weights are available for download at https://github.com/bonlime/keras-deeplab-v3-plus/releases/download/1.1/deeplabv3_mobilenetv2_tf_dim_ordering_tf_kernels.h5

In training, a batch size of 8 is used, meaning 8 images are read and augmented, and put together into an 8-by-257-by-257-by-13 array, with an 8-by-257-by-257-by-3 array for the mask (the last dimension is 2 in the case of a model trained only on cloud vs. non-cloud).

After a batch is processed by the model, the output predictions are compared to the true mask using a loss function, which defines how wrong the model's predictions were. We use categorical cross-entropy as our loss function, which is commonly used for machine learning. The categorical cross-entropy is defined:

$$CCE = -\sum_{i}^{C} \boldsymbol{t_i} \log(\boldsymbol{y_i}) \qquad [1]$$

Where t_i is the binary truth vector for the ith class, and y_i is the predicted confidence for that class, between 0->1. The gradients of this loss with respect to the model's weights are then calculated with back-propagation [28], and updated accordingly. The learning rate is initially set at 1e-3, with a momentum of 0.9, meaning the weights update accordingly (in this equation, i denotes the training step, rather than the position of the weight in a matrix):

$$w_{i+1} = w_i - \frac{\partial CCE}{\partial w_i} * (1 - momentum) * lr + (w_i - w_{i-1}) * momentum \qquad [2]$$

The first term in this equation is the weight before being updated, w_i, the second term is based on the gradient of the loss, and the third indicates the direction of travel of the weight in the update prior to this one. By using this momentum term, weight updates are smoothed as different examples are used in training, resulting in more reliable convergence.

Every 1,000 steps, the loss of the model on the validation set is calculated. After 100 epochs, the models had converged, at which point the learning rate was lowered by a factor of 10 (to 1e-4), and training resumed for another 100 epochs. Then, once the validation loss had converged again, training was stopped. Therefore, in total, the model was trained for 200,000 steps; with a batch size of 8, the model was trained on 1.6 million samples drawn from the training dataset. Training time was roughly 5-6 hours on an Nvidia RTX 2080 Super 8GB graphics card.

2.4.6 Resampling and Normalisation

Before being inputted into DeepLab v3+, all bands are resampled to 20 m/pixel, using bilinear interpolation. This is because the architecture of DeepLab v3+ is such that data from the different channels must have the same spatial extent. 20 m was chosen as this is the resolution of the other components in the HR-AlbedoMap system, and was also the same resolution as the labelled dataset. Sentinel-2 bands which are already delivered at 20 m/pixel (bands 5, 6, 7, 8a, 11 and 12) are unchanged, whilst those at 10 m/pixel (bands 2, 3, 4 and 8) are downsampled by a factor of two, and those at 60 m/pixel (bands 1, 9 and 10) are upsampled by a factor of 3.

Reflectance values are normalised from the integer values stored in the L1C products, by dividing the values by 10,000 and storing them as floating point numbers. The resampled and normalised bands are stored in memory as numpy arrays. If used with a full Sentinel-2 L1C product, this is an array of dimensions 5490-by-5490-by-13.

2.4.7 Sliding window

The *SlidingWindow* is a python class which feeds the model cropped sections of the image, and then stitches the outputs together into a final scene-wide cloud mask. Figure 4 shows the geometry and labelled dimensions of the sliding window. We trained DeepLab v3+ with a window size of 257-by-257 pixels. Therefore, we keep X_{window} at 257 pixels across, nevertheless, the calculation of N_X and X_{offset} are generally applicable. N_X is found with the following equation, where the floor operator rounds down to the nearest integer:

$$N_X = \boldsymbol{floor}\left(\frac{X_{scene} - X_{window}}{X_{stride}}\right) \quad [3]$$

With N_X we can now calculate the offset along the x dimension:

$$X_{offset} = \boldsymbol{floor}\left(\frac{N_X X_{stride} + X_{window} - X_{scene}}{2}\right) \quad [4]$$

These calculations are identical for the y-dimension. With these values, the corners of all windows can be calculated for indices *n* between *0->N_X*, and *m* between 0->N_Y:

$$X_{left}^{(n)} = -X_{offset} + nX_{stride}$$

$$X_{right}^{(n)} = -X_{offset} + nX_{stride} + X_{window}$$

$$Y_{top}^{(m)} = -Y_{offset} + mY_{stride}$$

$$Y_{bottom}^{(m)} = -Y_{offset} + mY_{stride} + Y_{window}$$

[5]

These define the coordinates for the *m*n* windows which are extracted from the scene and passed to the model. The bounding boxes are made into a list *m*n* long, and by inference they are taken in batches to be given to the model. The batch size can be set by the user and should be set based on the available memory on either the CPU or GPU being used.

For windows that are at the image's boundary, containing some areas with no-data, these values must be filled with something. DeepLab v3+ is a convolutional model, so the values in these regions can affect the classification of those within the image, and sharp boundaries may cause issues for the model. Therefore, we pad the image with the mean values for each band from within the valid section of the image. This ensures that there is not a sharp discontinuity at the boundary, by roughly matching the spectral profile of the valid data within the window, minimising the edge's effect on performance. For full Sentinel-2 L1C products at 20m/pixel, an X_{stride} of 238 leads to a very small value for X_{offset} of 2, resulting in minimal edge effects.

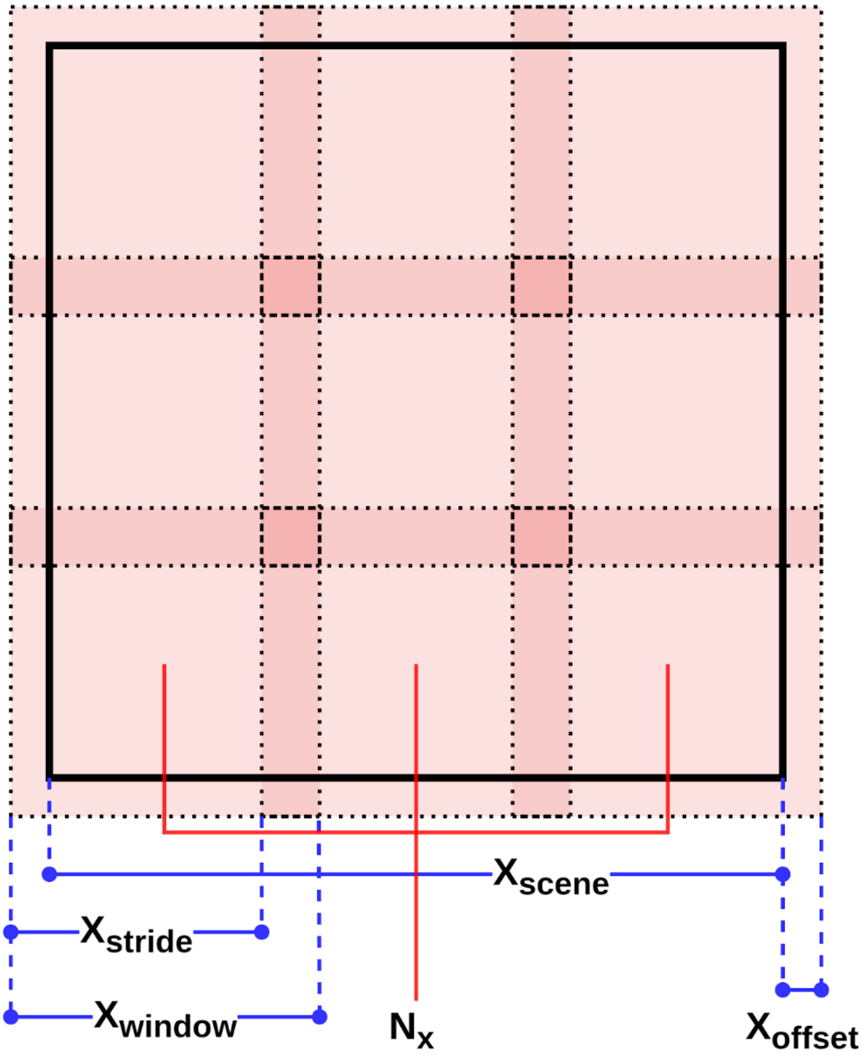


Figure 4. Sliding Window geometry for cloud mask. The model used defines Xwindow, whilst the user defined Xstride. Using these values, NX can be calculated, and then Xoffset.

2.4.8 Mask Aggregation

The outputs of each cropped window overlap, when X_{stride} and X_{window} are not equal, and so a scheme for combination of the masks must be made. We take the mean value at each given pixel. This means some areas with more overlaps receive more predictions, but the resulting confidence values after aggregation are still on the scale of 0-->1. For final class predictions, the class confidence values are thresholded at a value of 0.5. Then, for the clear vs. cloud+cloud-shadow model, the union between cloud and cloud-shadow pixels is taken.

2.4.9 Model Performance

Once trained, the two variations of DeepLab v3+ (the cloud vs non-cloud model and the clear vs. cloud/cloud-shadow) are tested on the data not used during training, in order to ascertain their performance on unseen data. Four metrics are used, defined with respect to 4 values: The *True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN)*. For both models, we take the "positive" class to be the cloud (and shadow, if included), whilst the clear pixels are treated as "negative". The four metrics are defined below:

$$Overall\ Accuracy = OA = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F_1 = \frac{2 * Recall * Precision}{Recall + Precision}$$

[6]

The model outputs a confidence value for each pixel, which we threshold at 0.5 for the purposes of the evaluation, so as not to bias results favourably by selecting an optimal threshold. Table 3 below shows the results of the two models on our test set, and some comparisons with other models, with our models in bold font.

Table 3: Performance results for 4 models. The first three rows show models tested for cloud vs. non-cloud, whilst the final row is a clear vs. cloud/shadow model. Those in bold are the ones developed in this project.

Model	Scenes	*OA*	*Recall*	*Precision*	F_1
s2cloudless (Zupanc, 2017)	256	92.4	89.5	95.9	92.6
Standard L1C mask	256	83.1	80.3	86.9	83.5
Cloud vs. non-cloud	**256**	**94.0**	**94.2**	**94.2**	**94.4**
Clear vs. cloud+shadow	**212**	**96.1**	**95.6**	**97.3**	**96.4**

We also conducted some tests over test sites and visualised their results. Figure 5 and Figure 6 show two examples of the clear vs. cloud/shadow model, demonstrating its capabilities on two very different areas, one in the USA containing high and low-level clouds and the other in Germany which show persistent clouds throughout the winter respectively.

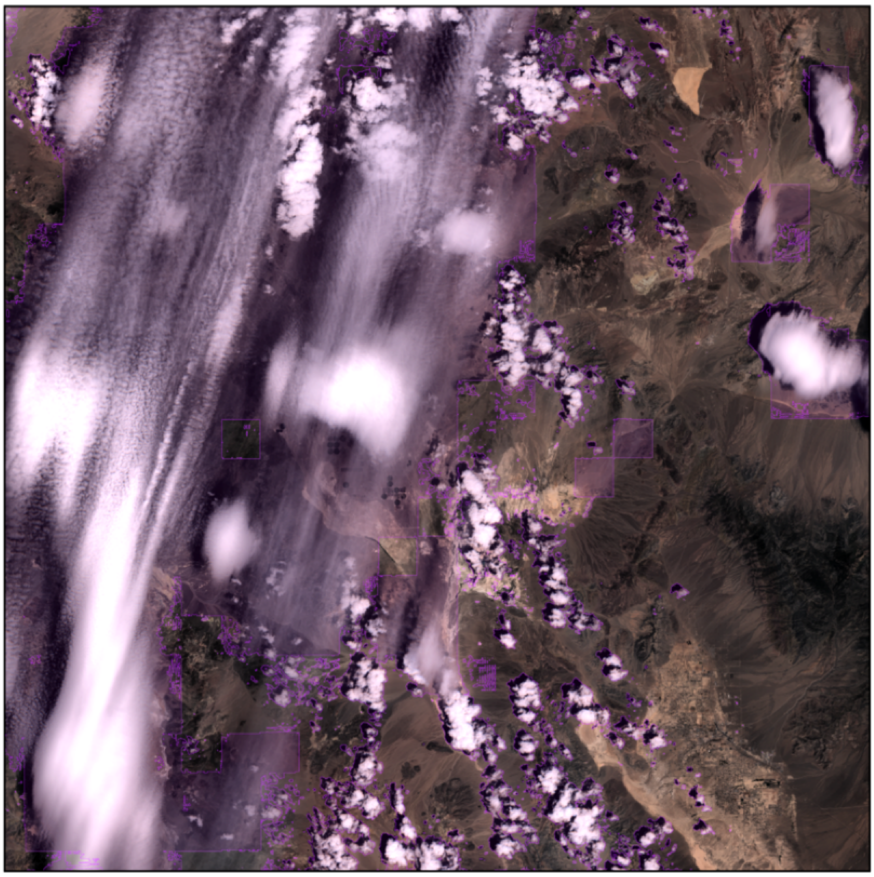


Figure 5. Clear vs. cloud/cloud-shadow for L1C product over Desert Rock, Arizona, from May 2020. Cloud/cloud-shadow boundaries are marked automatically and shown here in purple. Both thick and thin clouds are successfully detected, with few regions incorrectly flagged as non-clear. Most shadows are picked up successfully too.

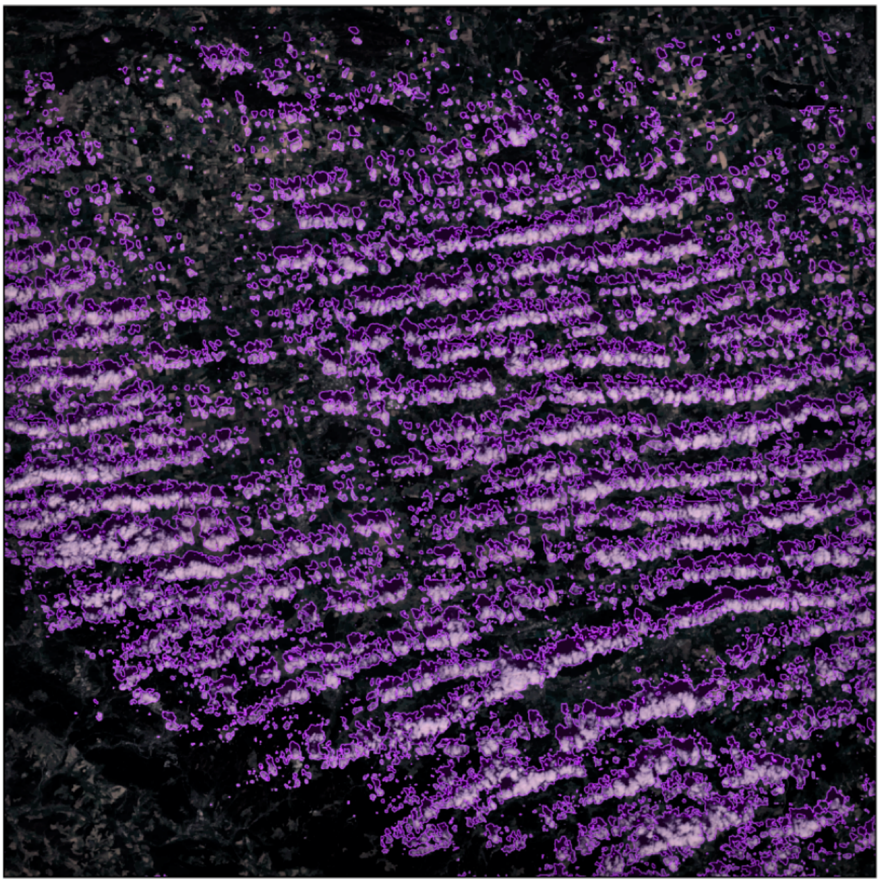


Figure 6. Clear vs. cloud/cloud-shadow results over Hainich, Germany. Shadows are picked up successfully even against dark forested terrain.

2.4.10 Error Budget Estimates

The errors in the cloud+cloud-shadow mask can be thought of as comprising two populations. Errors of omission, or False Negative Rate (those cloudy or cloud shadow pixels which are incorrectly taken as clear) and errors of commission, or False Positive Rate (those clear pixels which are taken to be cloudy or shadowed).

$$Omission = FNR = \frac{FN}{TP+FN} \qquad [7]$$

$$Commission = FPR = \frac{FP}{TN+FP} \qquad [8]$$

For our clear vs. cloud+cloud-shadow model's results on our test set, we achieve an *FNR* of **3.35%** and an *FPR* of **4.42%**. This means that 3.35% of cloudy or shadowed pixels will be incorrectly used, and 4.42% of clear pixels that could have been used will be missed.

2.5 Atmospheric Correction

This section introduces the SIAC atmospheric correction method [33], which is used to produce atmospherically corrected Sentinel-2 spectral surface reflectance. The basic idea of the SIAC method is to use a spectral BRDF dataset to describe the land surface anisotropy as a prior estimate of the Bottom-of-Atmosphere/Canopy (BoA) along with a coarse resolution dataset of atmospheric aerosol scattering and water vapour absorption in order to solve the inverse problem of retrieving the BoA.

The main steps of the SIAC method can be summarised as following:

a) Use the 500m daily MODIS BRDF product (known as MCD43A1) help to determine the spectral BRF given the Sentinel-2 acquisition geometry, in order to provide an expectation of the surface reflectance calculated at coarse resolution (500 m) and in the MODIS BRDF bands.

b) Using a set of linear transformations to convert the predicted reflectance from the previous step to S2 spectral bands. This yields an expectation of surface reflectance for the target geometry and spectral bands at coarse resolution.

c) Assuming that the surface reflectances are strongly correlated with the TOA reflectances, calculate an empirical Point Spread Function (PSF) model by maximising the correlation between the TOA reflectance convolved with a Gaussian PSF and surface reflectance coarse resolution expectation from step b).

d) Map the surface reflectances to the TOA using a radiative transfer model and estimates of atmospheric composition. These can be compared with the measured target sensor TOA reflectances convolved with the empirical PSF.

e) Exploiting the Copernicus Atmospheric Monitoring Service (CAMS) data as an *a priori* estimate, an inverse solution can be constructed to retrieve Aerosol Optical Thickness (AOT), Total Columnar Water Vapour (TCWV) and Total Columnar Ozone (TCO3). Furthermore, a spatial regularisation is used under the assumption of smooth variation of the atmospheric composition parameters.

f) The previous steps result in a complete inference of the *a posteriori* joint Probability Density Function (PDF) of the atmospheric parameters, which can then be used to correct the original TOA reflectance data using in our case a Lambertian surface-atmosphere coupling assumption.

g) Figure 7 shows the processing chain of the SIAC atmospheric correction method.

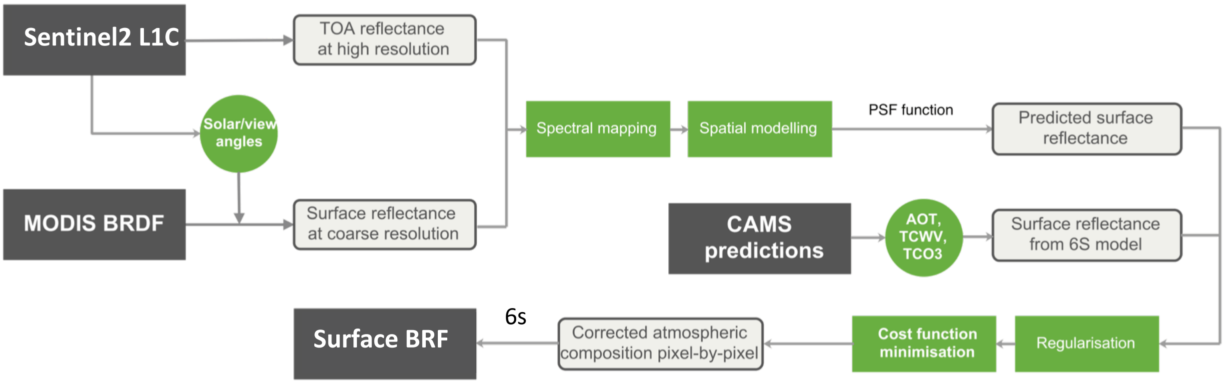


Figure 7. Flow Diagram of the SIAC processing chain.

2.6 Endmember Extraction and Spectral Unmixing

In this HR-Albedo framework, endmember extraction and spectral unmixing techniques are used as an unsupervised image classification method. This method employs endmember extraction to define pure pixel spectrum prior to a spectral unmixing stage. So-called spectral endmembers are often used as reference data in image classification. An endmember is defined as a land “type” that is assumed to have a unique spectral signature. Based on the extracted endmember, spectral unmixing provides the computation of the fractional contribution of individual endmembers within each pixel. The steps of employing the endmember extraction and spectral unmixing over a Sentinel-2 tile (110 km by 110 km) are shown in Figure 8, which includes the following major steps:

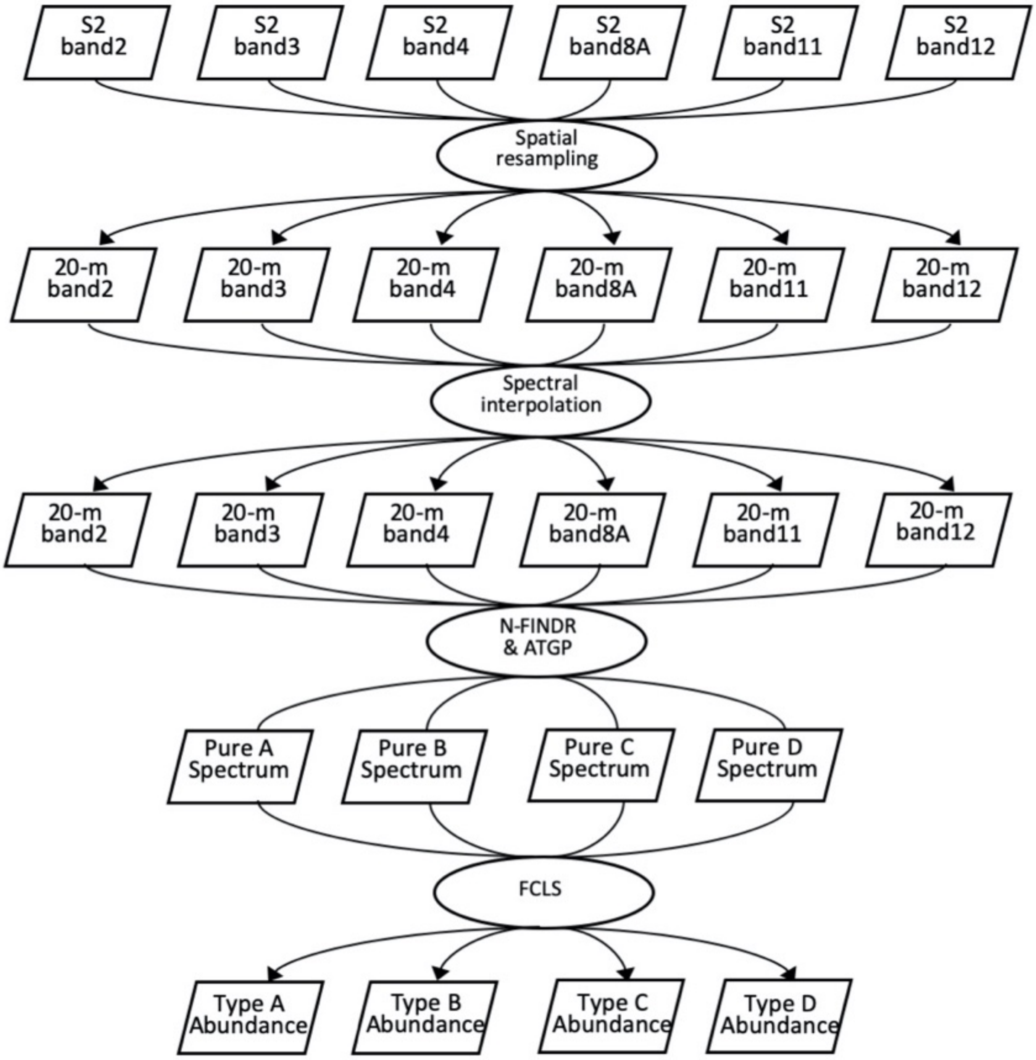


Figure 8. Steps of employing endmember extraction and spectral unmixing.

h) Spectral band selection.

The selected input spectral bands include the following 6 bands: band-2 (492.4 nm), band-3 (559.8 nm), band-4 (664.6 nm), band-8A (864.7 nm), band-11 (1613 nm) and band 12 (2202 nm). These spectral bands are chosen as they are very close to the corresponding MODIS and VIIRS spectral bands.

i) Spatial resampling.

The S2 band-8A, 11 and 12 are 20-m resolution data, and band-2, 3 and 4 are 10-m resolution data. The generated cloud mask data is also at 20-m resolution. In this step, all spectral data are resampled to 20-m resolutions. Specifically, 4 pixels at 10-m resolution are averaged into a 20-m resolution pixel.

j) Spectral interpolation.

The endmember extraction method was originally proposed for hyperspectral data, where a large amount of spectral reflectance values is available. In this HR-Albedo study, it is found that the endmember extraction can be also applied to multispectral S2 data. However, the performance of extracting endmembers is improved with increased information (details not shown here) in the spectral domain. Therefore, in this step a spectral interpolation is used to increase the spectral sampling between 490 and 860 nm.

k) Endmember extraction using N-FINDR and ATGP.

N-FINDR algorithm [29], one of the most popular and effective endmember extraction algorithms, starts with the random initialization of the endmember set. In this framework, N-FINDR is used to extract endmembers from 20-m interpolated S2 surface reflectance data. To improve the performance of endmember extraction, an Automatic Target Generation Process (ATGP) algorithm is used to initialize the endmember set in the endmember search process. Figure 9 shows an example of the extracted endmembers from S2 tile 32UNB that covers the FLUXNET tower site in Hainich. This example shows that 4 endmembers are identified and extracted from the spatially mixed pixels. The endmember type-D clearly represents vegetation, and the endmember type-B&C represent different types of soil. The endmember type-A with very high spectral reflectance values represent bright features (e.g. buildings or isolated clouds missed previously) in the data.

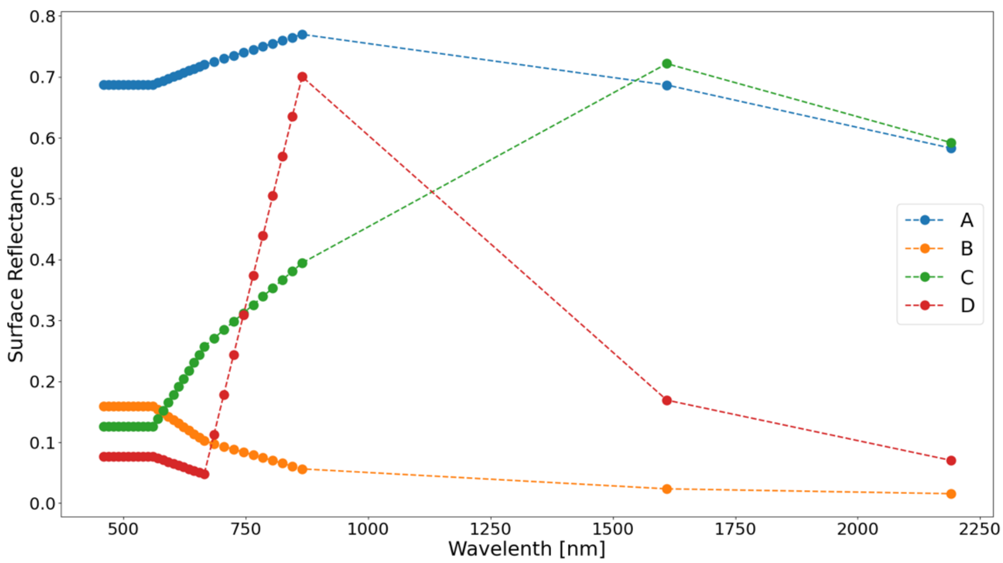
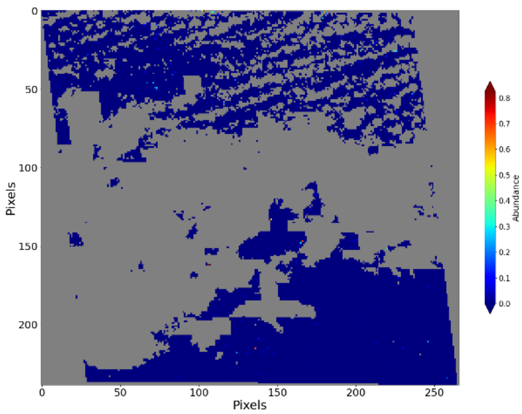


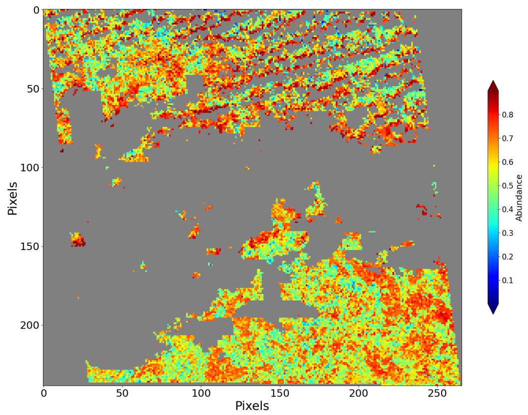
Figure 9. Example of extracted endmembers from S2 tile 32UNB on 25th July 2019.

l) Spectral unmixing.

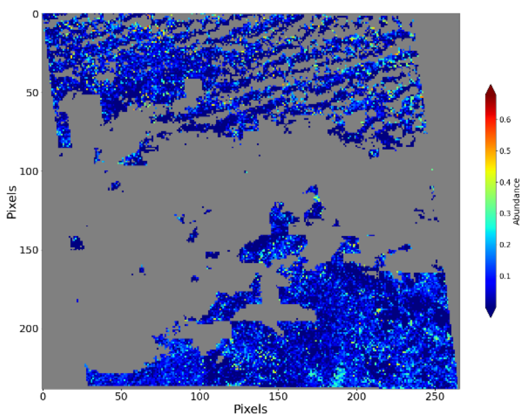
The aim of employ spectral unmixing is to compute the fractional contribution of individual endmembers. In this module, a Fully Constrained Least Squares (FCLS) [30] linear unmixing method is used. Figure 10 shows an example of different abundance maps that are generated for individual endmembers. It can be observed that type-A endmember has very low abundance values as there are very few bright features (e.g. buildings) in this area. Type-B endmember represents the soil and has relatively high abundance ratios that are distributed across the whole area. Type-C endmember represents another type of soil and has lower abundance ratios than type-B soil. Type-D represents the vegetation and it has abundance ratios that can reach 0.7 in some regions. In this example, the displayed abundance maps are at 500-m Sinusoidal projection. This means the 20-m S2 data are first resampled to 500-m resolution, and then reprojected to MODIS Sinusoidal projection system as these abundance ratios at 500-m Sinusoidal projection will be used as input data in the next stage for HR-Albedo retrievals.



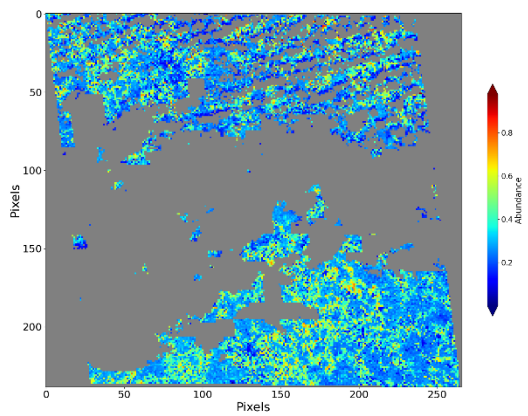
(a) Abundance map for type A



(b) Abundance map for type B



(c) Abundance map for type C



(d) Abundance map for type D

Figure 10. Example of generated abundance map for individual endmembers.

2.7 High-Resolution Albedo (HR-Albedo) Retrieval

The basic idea of retrieving high-resolution spectral and broadband albedo follows the method that was proposed in [31]. This method builds a forward albedo-to-reflectance model at 500-m MODIS Sinusoidal projection, and uses this model to invert the HR-albedo at S2 20-m resolutions. The retrieval module consists of the following steps:

a) Estimation of 500-m albedo at Sinusoidal projection.

The MODIS 500-m Bi-directional Reflectance Distribution Function (BRDF) data are identified as the MCD43A1 products. The VIIRS 1-km BRDF data are identified as the VNP43MA1 products. The surface reflectances derived from this kernel-driven BRDF model are described as:

$$R(\lambda, \theta_{in}, \theta_{out}, \phi) = f_{iso}(\lambda) + f_{vol}(\lambda)k_{vol}(\theta_{in}, \theta_{out}, \phi) + f_{geo}(\lambda)k_{geo}(\theta_{in}, \theta_{out}, \phi) \quad [9]$$

where λ is the band-centre of a given spectral channel; θ_{in}, θ_{out} and ϕ are the solar zenith, view zenith and relative azimuth angles, respectively. k is the BRDF RossThick-LiSparse-Reciprocal (RTLSR) kernel and f is the spectrally-dependent kernel weighting, with subscripts *iso*, *vol* and *geo* representing the isotropic, volumetric and geometric-optical components, respectively. Integration of the BRFs over all view angles results in a DHR, and a further integration over all illumination angles results in a BHR:

$$DHR_M(\theta_{in}) = \frac{1}{\pi}\int_0^{2\pi} d\varphi \int_0^1 R_M(\lambda, \theta_{in}, \theta_{out}, \varphi) u_v du_v \quad [10]$$

$$BHR_M = \frac{1}{\pi}\int_0^{2\pi} d\varphi \int_0^1 DHR_M(\theta_{in}) u_s du_s \quad [11]$$

where u_v(=$\sin\theta_{out}$) and u_s(=$\sin\theta_{in}$) are the variables of integration. Alternatively, the DHR and BHR can be estimated using a simple polynomial with good accuracy for any solar zenith angle. The polynomial is as follows:

$$\begin{aligned} DHR_M(\theta_{in}) = \quad & f_{iso}(\lambda)\left(g_{0_{iso}} + g_{1_{iso}}{\theta_{in}}^2 + g_{2_{iso}}{\theta_{in}}^3\right) + \\ & f_{vol}(\lambda)\left(g_{0_{vol}} + g_{1_{vol}}{\theta_{in}}^2 + g_{2_{vol}}{\theta_{in}}^3\right) + \\ & f_{geo}(\lambda)\left(g_{0_{geo}} + g_{1_{geo}}{\theta_{in}}^2 + g_{2_{geo}}{\theta_{in}}^3\right) \end{aligned} \quad [12]$$

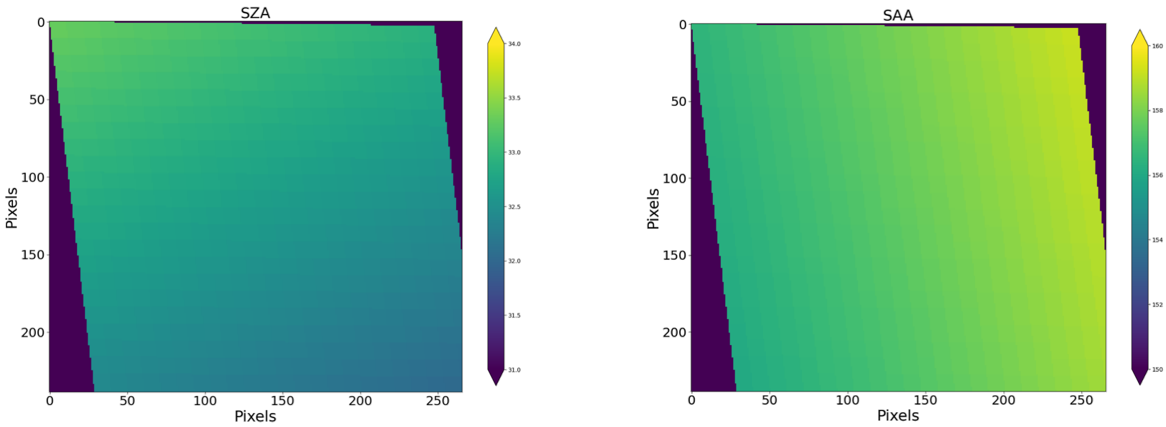
$$BHR_M = f_{iso}(\lambda)g_{iso} + f_{vol}(\lambda)g_{vol} + f_{geo}(\lambda)g_{geo} \quad [13]$$

The polynomial coefficients [32] for estimating DHRs and BHRs are included in Table 4.

Table 4. Polynomial coefficients for estimating DHRs and BHRs.

Term	Isotropic (iso)	RossThick (vol)	LiSparseR (geo)
g_0	1.0	-0.007574	-1.284909
g_1	0.0	-0.070987	-0.166314
g_2	0.0	0.307588	0.041840
white-sky integral	1.0	0.189184	-1.377622

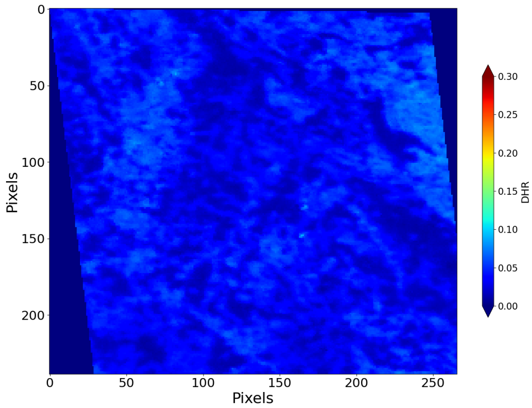
Following the polynomial in Eq. (12) & (13), the 500-m Sinusoidal projection DHR and BHR are calculated using MODIS BRDF, and solar/viewing geometry at S2 overpass time. Figure 11 shows an example of resampled S2 Solar Zenith Angle (SZA) and Solar Azimuth Angle (SAA) data at 500-m Sinusoidal projection.



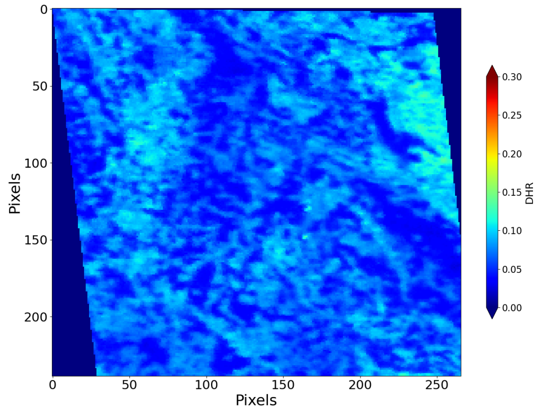
(a) aggregated S2 SZA (a) aggregated S2 SAA

Figure 11. Example of S2 Solar Zenith Angle (SZA) (a) and Solar Azimuth Angle (SAA) (b) that are resampled and aggregated to 500-m Sinusoidal projection.

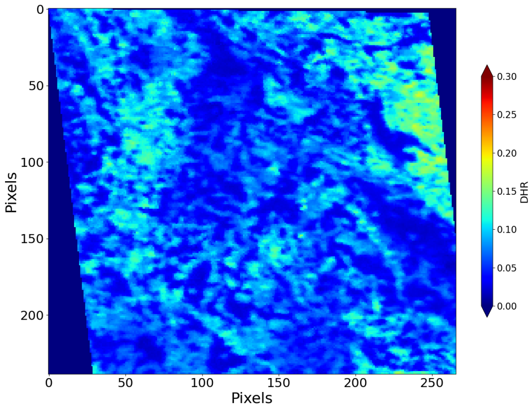
Based on the known solar angles and polynomial coefficients shown in Table 4, Figure 12 shows examples of calculated surface DHRs for the following MODIS bands: 470nm, 555nm, 645nm, 1640nm, 2130nm, VIS, NIR and SW bands.



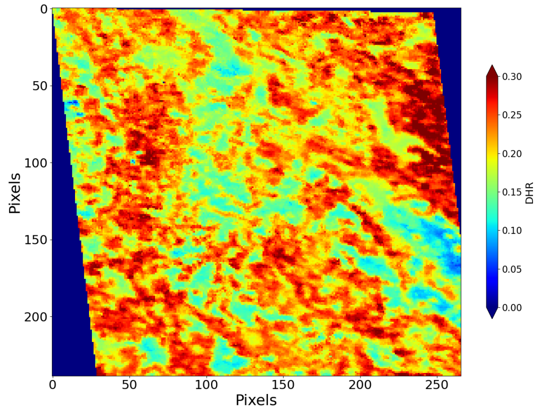
(a) 470nm DHR



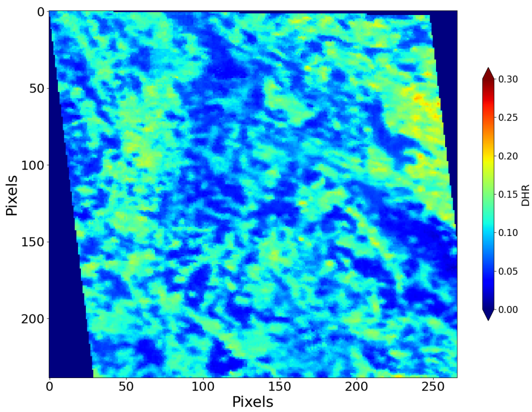
(b) 555nm DHR



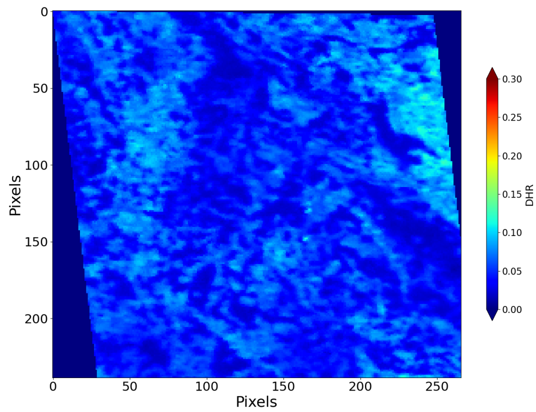
(c) 645nm DHR



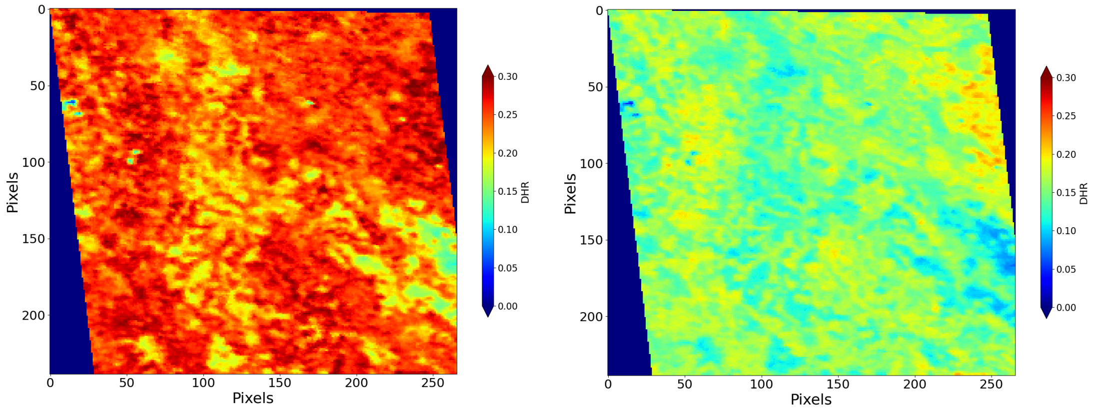
(d) 1640nm DHR



(e) 2130nm DHR



(f) VIS-band DHR

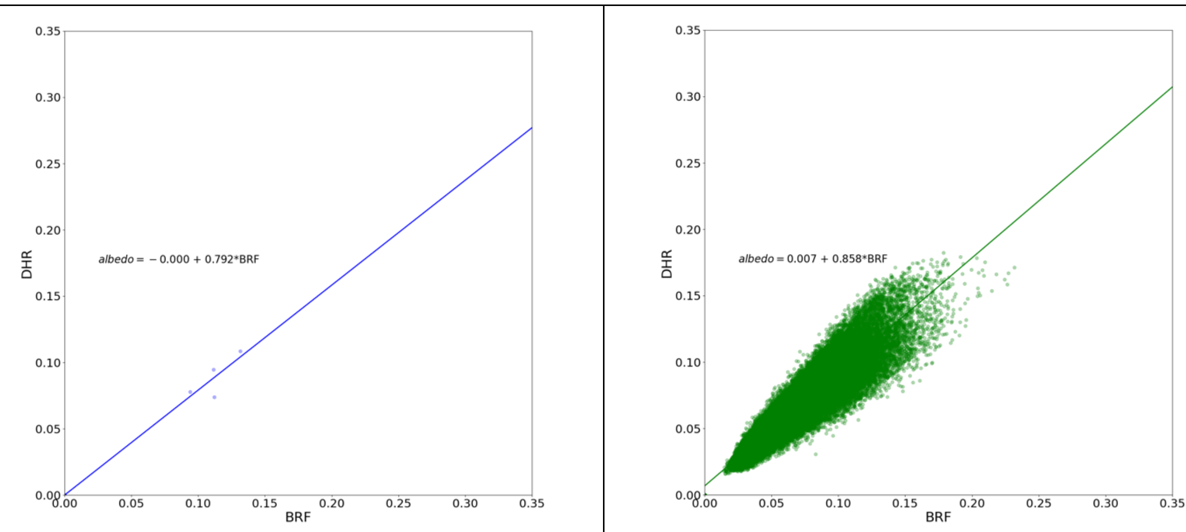


(g) NIR-band DHR (h) SW-band DHR

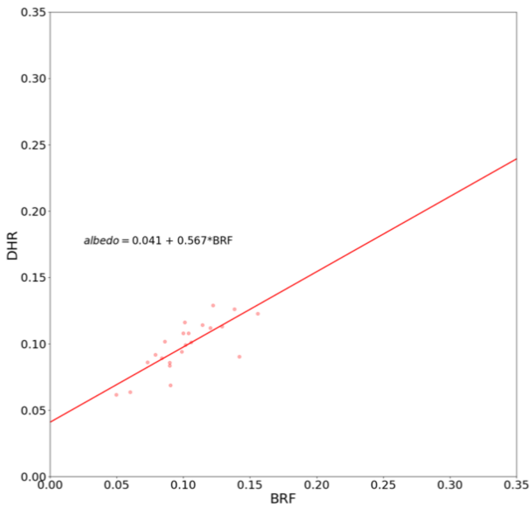
Figure 12. Example of surface DHRs that are estimated from the MODIS BRDF polynomials at 500m.

b) Forward modelling of albedo-to-reflectance regressions for individual endmembers.

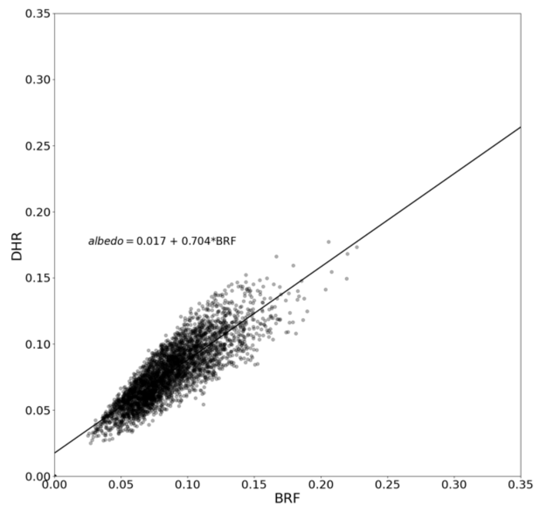
This step builds the forward model that describes the albedo-to-reflectance ratios for different endmembers. The ratios will be further applied to S2 spectral reflectances to retrieve high-resolution albedo values. A linear regression is used to predict the relations between albedo and reflectances for 8 bands (5 spectral bands and 3 broadbands) and 2 types of albedos (DHR and BHR), as shown in Figure 13.



(a) 645nm Type-A (b) 645nm Type-B



(c) 645nm Type-C



(d) 645nm Type-D

Figure 13. Example of regressions between DHRs and BRFs at 645nm band for different endmembers.

c) Apply the regression model to high-resolution S2 data.

In the step, the obtained regression models are applied to high-resolution S2 data to inverse the high-resolution albedo values. First, Table 5 lists the MODIS spectral bands that the forward regression models are based on, and the corresponding S2 bands that the retrieval applied to.

Table 5. List of MODIS bands and corresponding S2 bands.

MCD43A1 Bands	VIIRS Bands	Sentinel-2 Bands
Band-3 (470nm)	Band-M3 (483nm)	Band-2 (490nm)
Band-4 (555 nm)	Band-M4 (555nm)	Band-3 (560 nm)
Band-1 (645 nm)	Band-M5 (672nm)	Band-4 (665 nm)
Band-2 (859 nm)	Band-M7 (866nm)	Band-8A (865nm)
Band-6 (1640 nm)	Band-M10 (1610nm)	Band-11(1610 nm)
Band-7 (2130 nm)	Band-M11 (2255nm)	Band-12 (2190 nm)
Band VIS	Band VIS	Broadband VIS Eq. (15)
Band NIR	Band NIR	Broadband NIR Eq. (16)
Band SW	Band SW	Broadband SW Eq. (14)

Eq. (14)-(16) list the S2 narrow to broadband conversions, which are using coefficients proposed in [35].

$$b_{SW} = -0.0049 + 0.2688b_2 + 0.0362b_3 + 0.1501b_4 + 0.3045b_{8A}A + 0.1644b_{11} + 0.0356b_{12} \quad [14]$$

$$b_{VIS} = -0.0048 + 0.5673b_2 + 0.1407b_3 + 0.2359b_4 \quad [15]$$

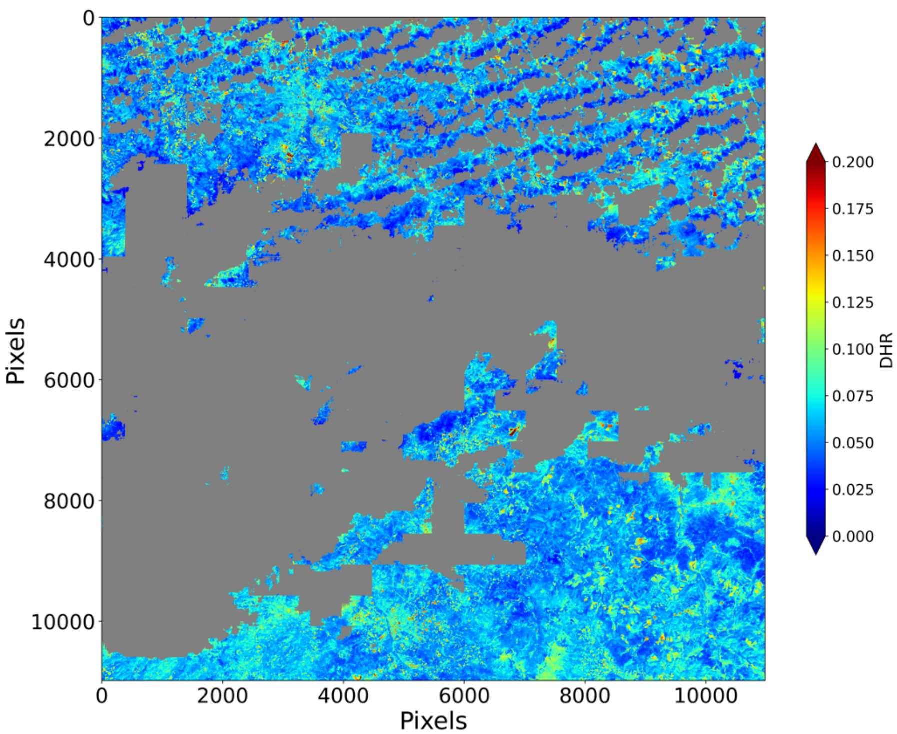
$$b_{NIR} = -0.0073 + 0.5595b_{8A}A + 0.3844b_{11} + 0.0290b_{12} \quad [16]$$

To generate high-resolution albedo at S2 resolutions, Eq. (17) is used to combine the regressions from the contributions of 4 different endmembers:

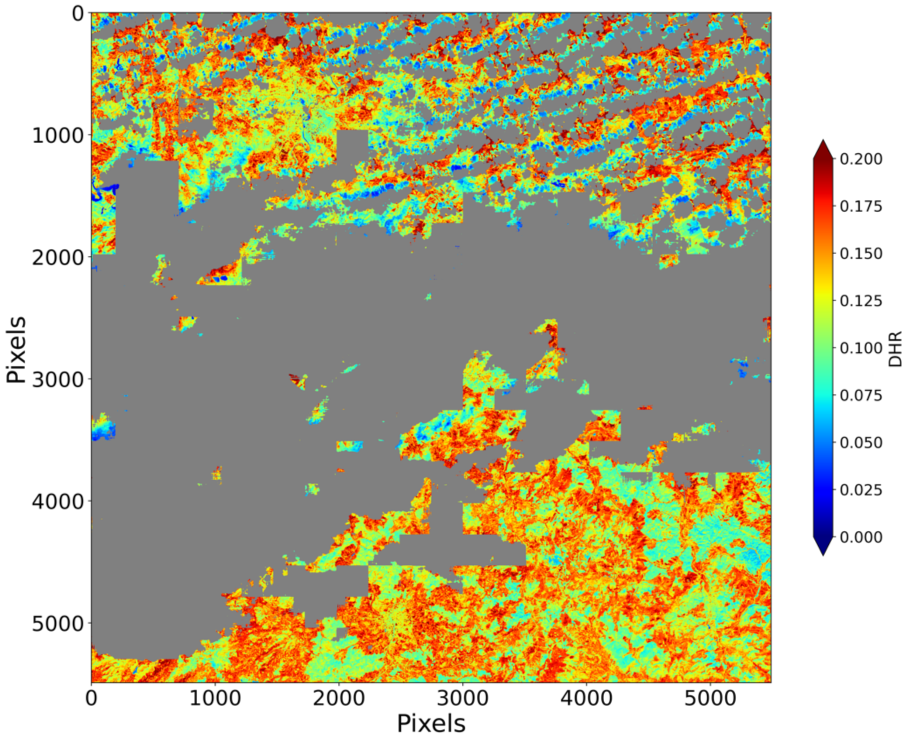
$$albedo(\lambda) = \sum_{i=1}^{N} w_i(c_i + k_i \times BRF(\lambda)) \quad [17]$$

where N is the number of endmembers, w_i is the abundance from endmember type *i*. c_i and k_i are the regression coefficients.

A full S2 tile has an extent of 110km * 110km. In this framework, the high-resolution albedo is retrieved in a patch of 1000 * 1000 pixels, with 50 pixels overlap between adjacent patches. The retrieval of individual patches is independent from each other, and can be processed in parallel or on multi-threaded CPU cores in a suitable supercomputing environment. Figure 14 shows an example of retrieved 10/20m albedo products.



(a) 10m DHR at S2 band-4 (665 nm)



(a) 20m DHR at broadband SW

Figure 14. Example of 10m/20 albedo products. Grey pixels are cloud covered.

2.8 Verification

The processed high-resolution albedo products have all been validated using tower-based albedometer or sun photometer measurements. The spectral albedo products have been validated using CNES RadCalNet [36] BHR spectral measurements, whilst the broadband shortwave albedo products are validated using GbOV (https://gbov.acri.fr/dataaccessLP/) measurements.

Figure 15 shows a flowchart for the validation of the S2 high-resolution albedo products. The high-resolution albedo pixels within the tower field-of-view are averaged to get a mean S2 high-resolution albedo value. The tower sun photometer measurements are processed by CNES through the ROSAS software to retrieve spectral albedo at the 2 RadCalNet sites, or the albedometer is processed by the GbOV LP2 processor to get broadband shortwave albedo at the GbOV sites.

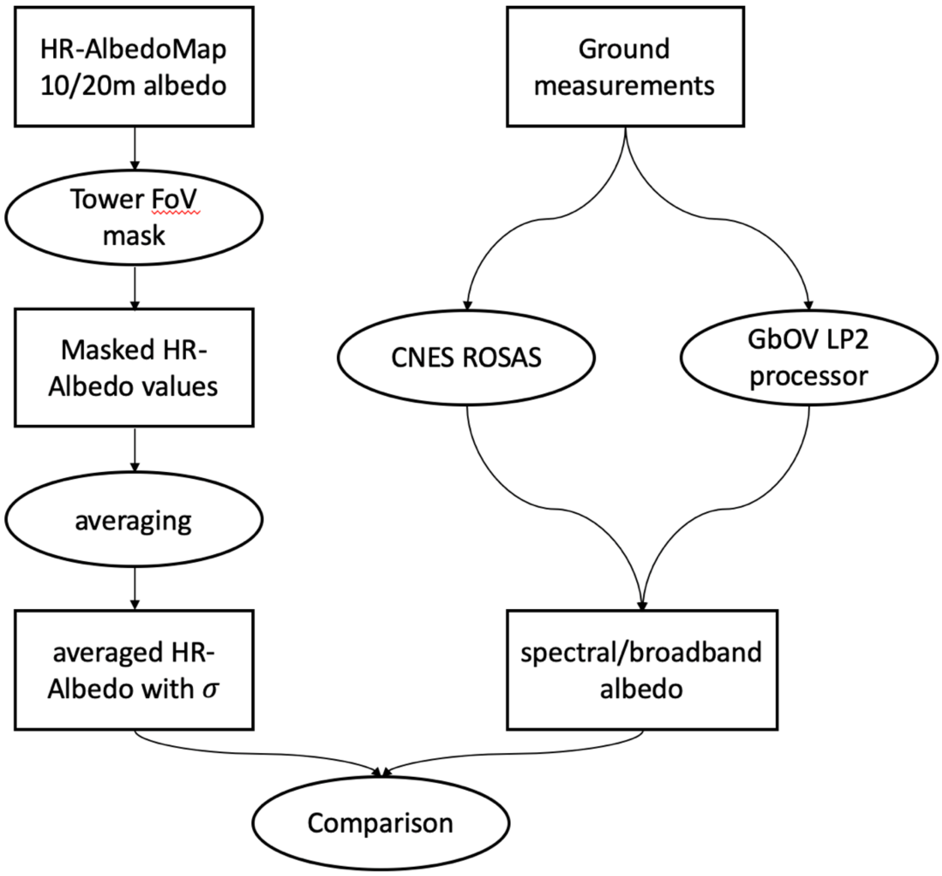
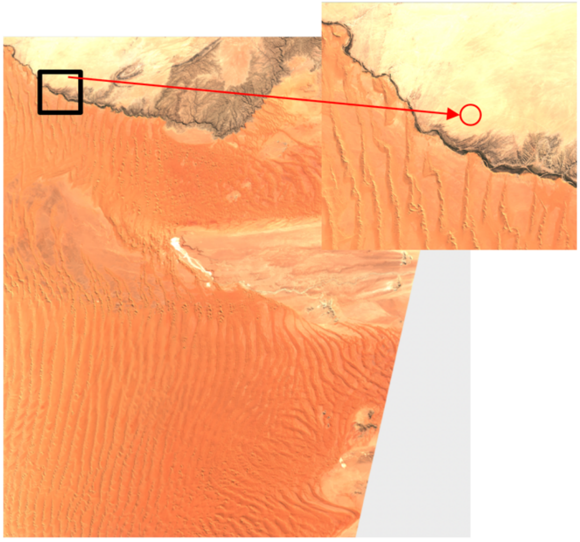


Figure 15. Flowchart of validating S2 high-resolution albedo products.

The verification of processed spectral high-resolution albedo products are performed using spectral RADCALNET *in situ* measurements kindly provided by Aimé Meygret and Morgan Farges from CNES Toulouse [37]. The two RADCALNET verification sites are Gobabeb (-23.6°S, 15.12°E) and LaCrau (43.56°N, 4.86°E). The geolocations and field-of-view of ground-truth measurement are shown in Figure 16.

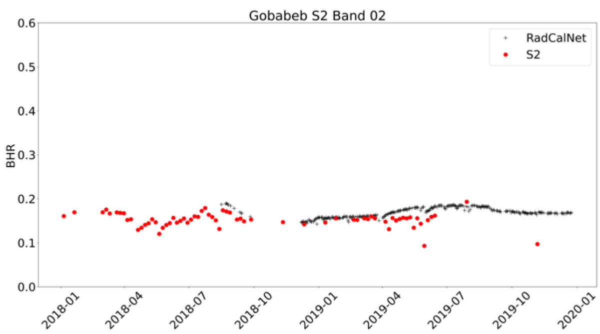
(a) 31TFJ Sentinel-2 Surface Reflectance RGB



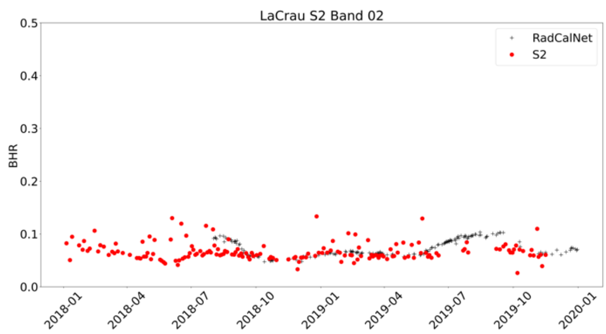
(b) 33KWP Sentinel-2 Surface Reflectance RGB

Figure 16. Geolocations and field-of-view of the towers from the RADCALNET sites (a) Gobabeb and (b) LaCrau. The red circle has a radius of 30 m.

CNES provided the BHR values that they calculated using the same way that the CNES RADCALNET surface reflectance values are derived [37]. In this validation stage, the Sentinel-2 band-2, band-3, band-4, band-8A and band-11 BHRs are compared between the RADCALNET measurements and HR-AlbedoMap products. No Band-12 is recorded by the CIMEL-318T used to obtain the sun photometer measurements of the sky and the ground.



(a)



(f)

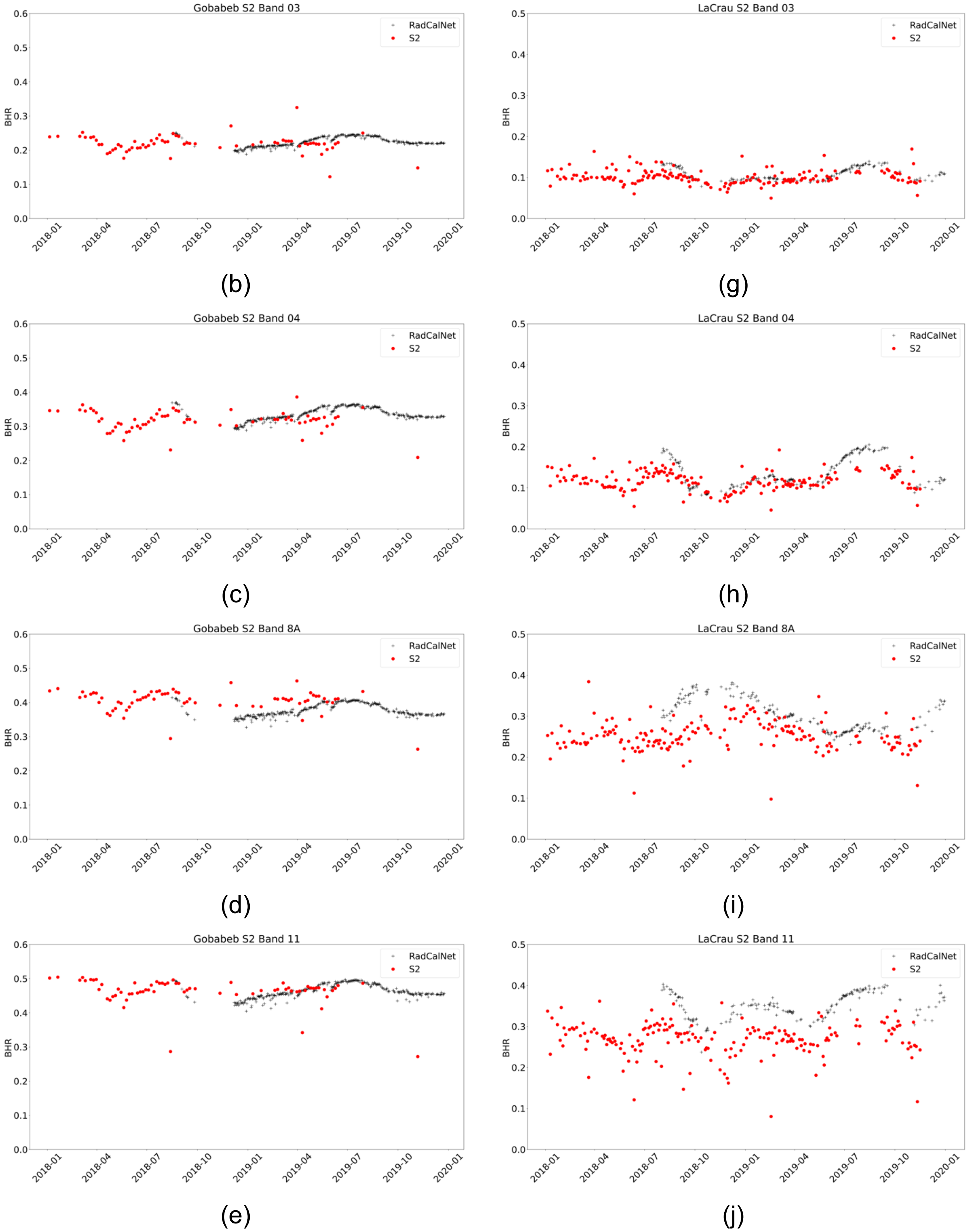
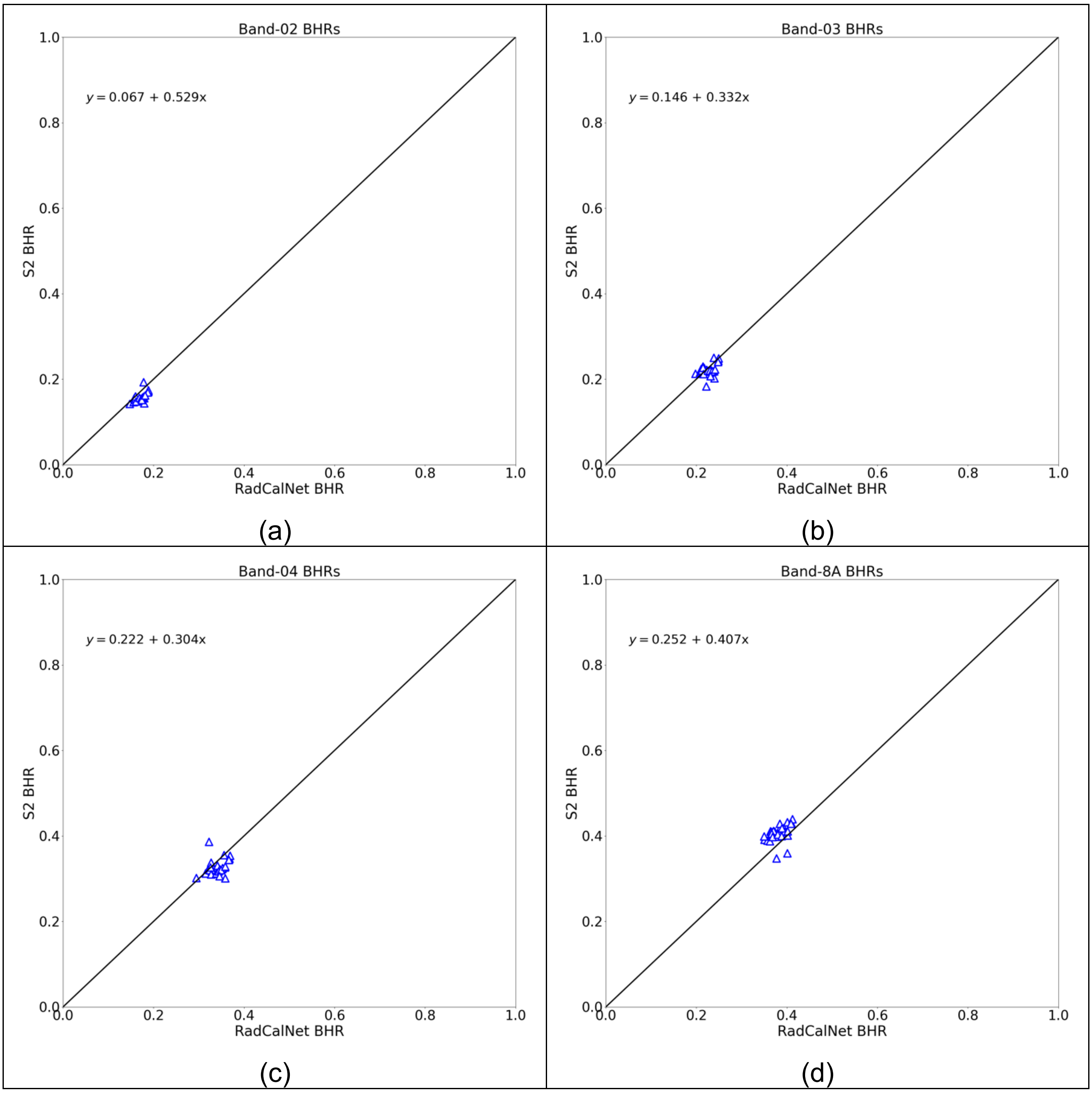
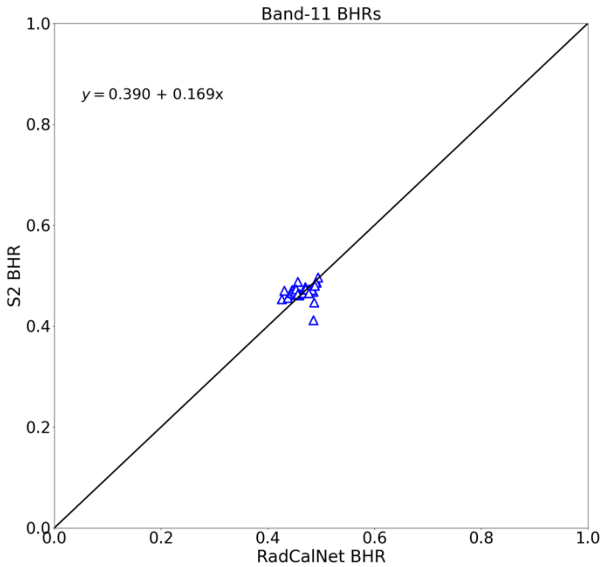


Figure 17. Intercomparison between RADCALNET BHRs (black dots) and HR-AlbedoMap retrieved BHRs (red dots) at band-2, band-3, band-4, band-8A band-11.

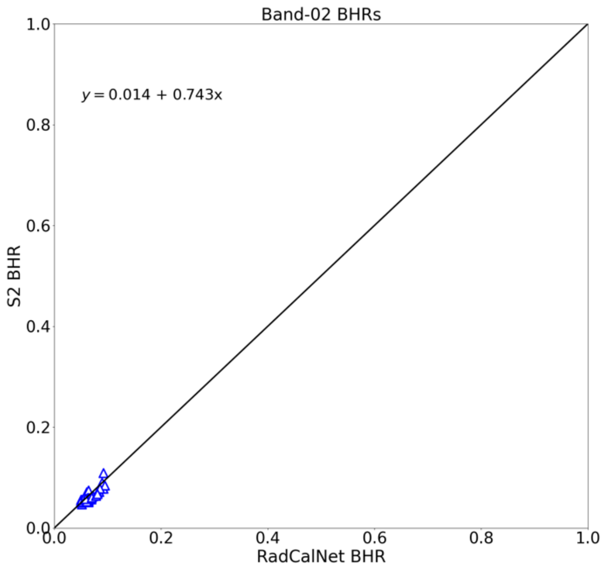
The corresponding 2D scatter plots for spectral BHRs between RADCALNET measurements and HR-AlebdoMap values are shown in Figure 18 for Gobabeb and Figure 19 for LaCrau. For both sites, band-2,3 and 4 display a good correlation between RADCALNET and HR-AlbedoMap retrievals. For bands 8A and 11, the HR-AlbedoMap BHRs at LaCrau site appear to be underestimated compared with RADCALNET measurements.



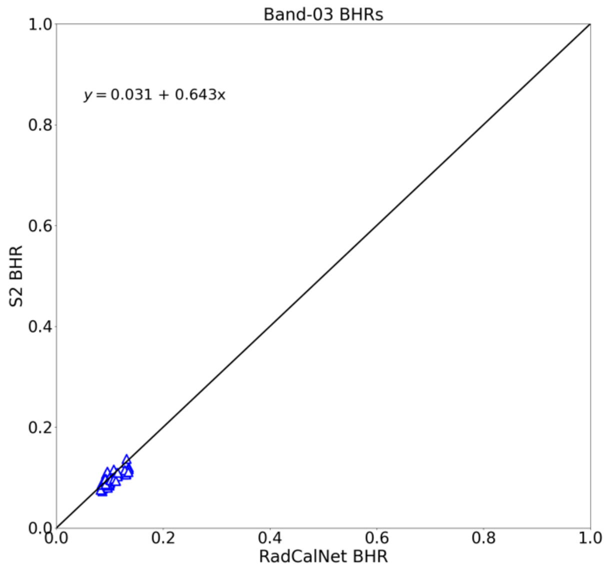


(e)

Figure 18. 2D scatter plots for BHR comparison between RADCALNET measurements and HR-AlbedoMap retrievals at Gobabeb.



(a)



(b)

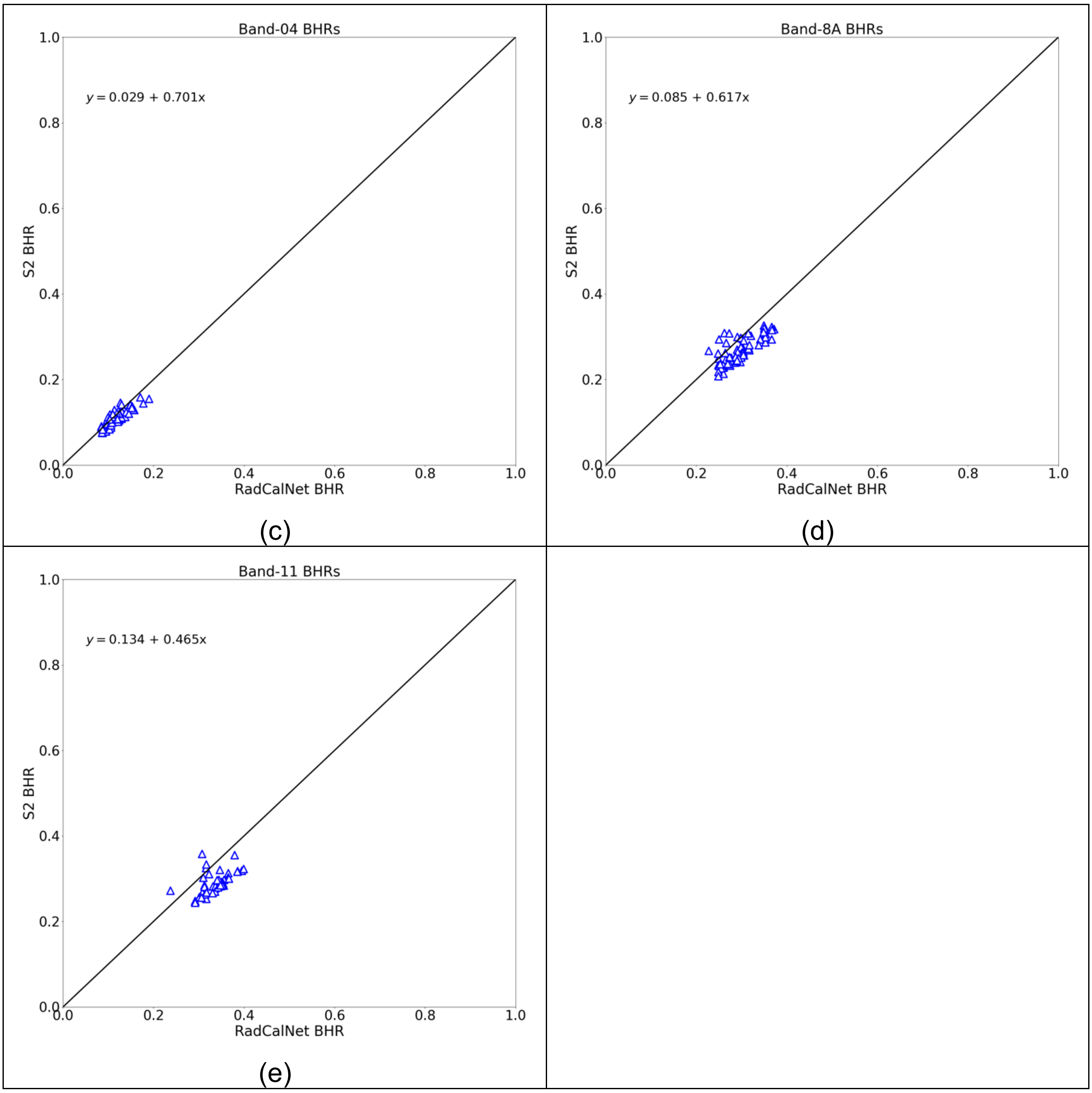
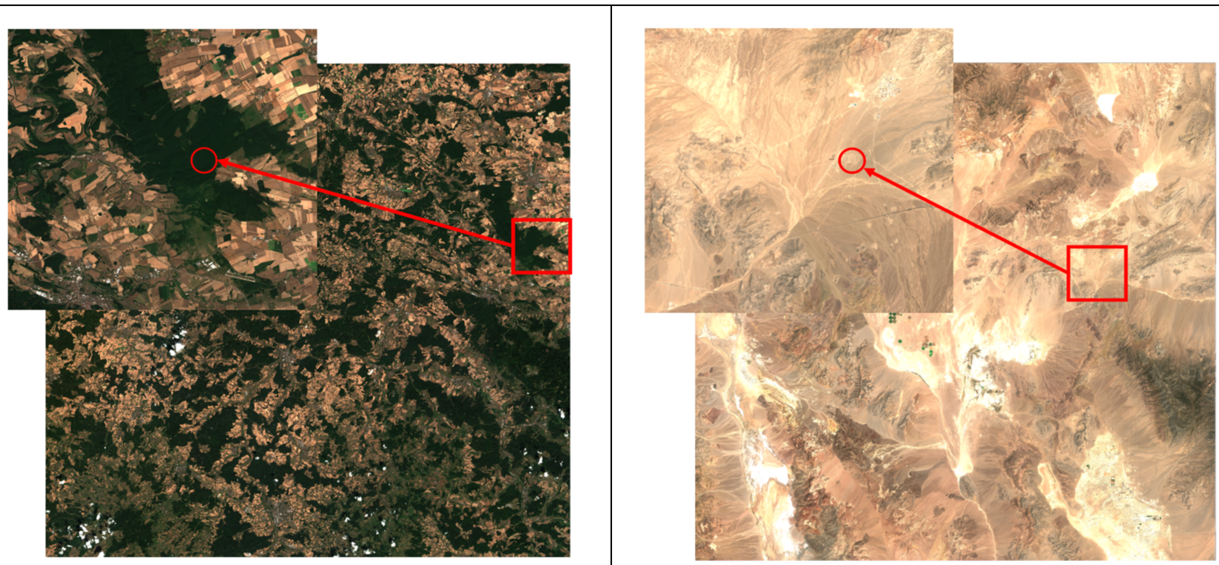


Figure 19. 2D scatter plots for BHR comparison between RADCALNET measurements and HR-AlbedoMap retrievals at LaCrau.

The processed high-resolution broadband shortwave albedo products have also been validated at two GbOV tower sites: Hainich (Germany) and Desert Rock (US). The GbOV sites provides broadband shortwave DHRs and BHRs that are derived from albedometers at over 20+ tower sites worldwide. The GbOV data are available from: https://gbov.acri.fr/dataaccessLP/. The methods of deriving GbOV tower-based DHRs and BHRs are described in [38]. In summary for the GbOV method, radiance data with a direct-to-diffuse ratio smaller than a threshold (β_{max}) are used to estimate the DHR, while data with this "diffuse ratio" larger than a threshold (β_{min}) are used to estimate the BHR.

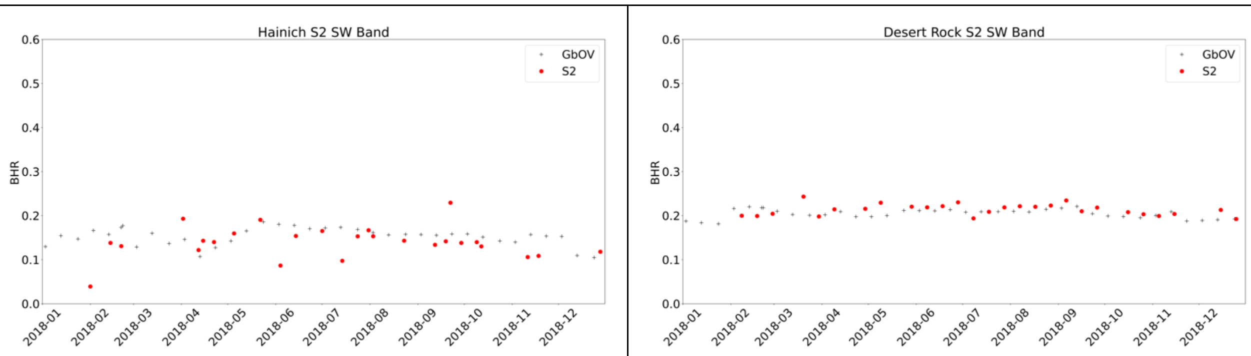


(a) 32UNB Sentinel-2 Surface Reflectance RGB

(b) 11SNA Sentinel-2 Surface Reflectance RGB

Figure 20. Geolocations and field-of-view of GbOV sites DE-HAI (a) and US-DRA (b). The red circle in the zoom window represents the tower field-of-view.

Figure 21 shows the broadband shortwave albedo intercomparison between GbOV ground measurements and HR-AlbedoMap retrievals.



(a) (b)

Figure 21. Intercomparison between GbOV broadband shortwave BHRs (black dots) and HR-AlbedoMap retrieved BHRs (red dots) at (a) Hainich (b) Desert Rock.

The corresponding 2D scatter plots for spectral BHRs between GbOV and HR-AlbedoMap retrievals are shown in Figure 22.

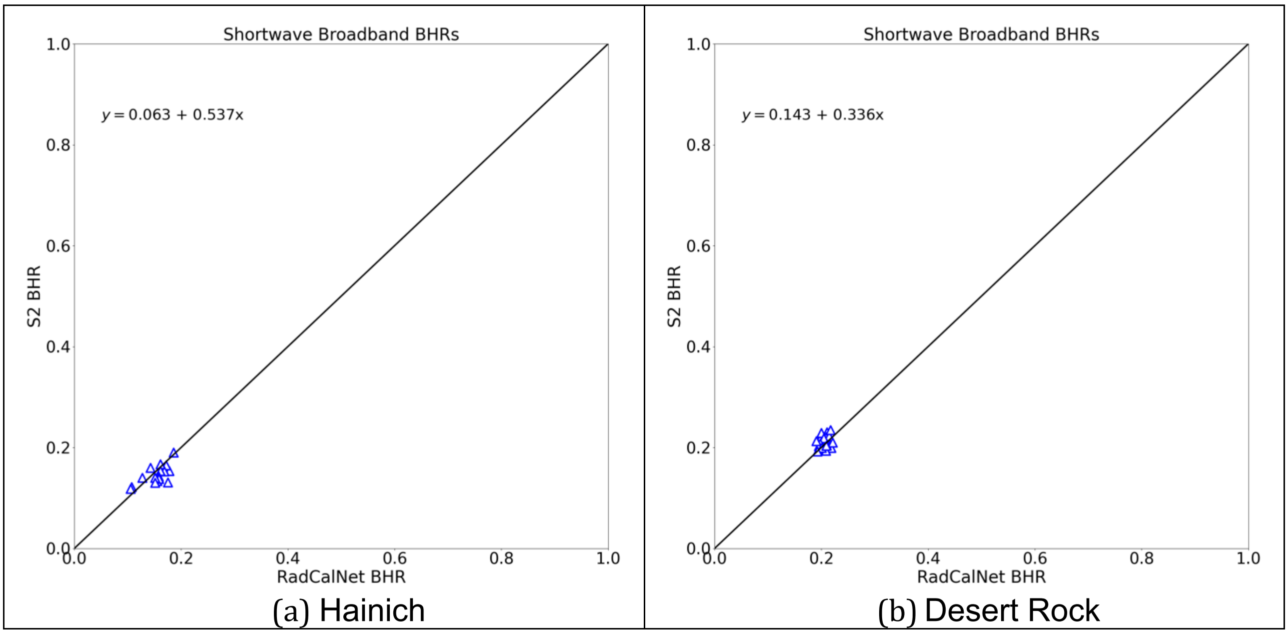


Figure 22. 2D scatter plots for broadband shortwave BHR comparison between GbOV measurements and HR-AlbedoMap retrievals at (a) Hainich and (b) Desert Rock.

2.9 Gap filling using a land surface BRDF climatology, the so-called "*prior*"

In order to generate a continuous spectral or broadband high resolution spectral and broadband albedo product, a background climatology needs to be employed for cloud-covered pixels. For cloud-covered pixels in high-resolution albedo maps, cloud-free high-resolution albedo values and MODIS (or VIIRS) BRDF *prior* from adjacent days are used for the interpolation. This window is set by number of cloud-free acquisitions per pixel and initially will be set to ±15 days. This section describes the method of generating the BRDF prior.

The MODIS Collection V6.1 (or VIIRS VNP43MA1) BRDF-Albedo model parameters product is used to develop the prior parameter estimates. Specifically, MDC43A1 (BRDF-Albedo Model Parameters 16-Day L3 Global 500m) and MCD43A2 (BRDF-Albedo Quality 16-Day L3 Global 500m) are used. Any optimal merging of data requires estimates of the uncertainties associated with the various data sources. Often this information is not directly available. The particular dataset of interest here is the MODIS (or VIIRS) BDRF/Albedo product (MCD43 or VNP43). BRDF or Albedo can be described for some waveband via spatial datasets of three model parameters f_{iso}, f_{vol}, and f_{geo}, available from the MCD43A1 or VNP43iA1 or VNP43MA1 product at 500m spatial resolution (MOD43A1 & VNP43i) or at 1000m for VNP43mA1 on daily time-steps from early 2000 to the present (i.e. around 21 years of data) for MODIS and from 2012 to present (i.e. around 9 years) for SNPP-VIIRS. The uncertainty in albedo is a function of the angular sampling achieved by the MODIS instruments and uncertainties in the input BRF data (due to errors in the correction of atmospheric effects, footprint/gridding issues etc.). If insufficient samples (≤7 are deemed available over the (16 day) period of observation, the linear BRDF model cannot be inverted without further constraint and a 'backup algorithm' is brought to bear on the problem. In this case, a constrained model is used in the inversion (the so-called *magnitude inversion*). For these various reasons there is no routine assessment of the uncertainty associated with the BRDF model parameters. Instead, to give the user a guide to the reliability of the data, the product MCD43A2 provides Quality Assurance(QA) information associated with each pixel and time period of inversion. For each of the first seven MODIS wavebands and the I and M bands of the VIIRS product, QA is categorised into five levels as shown in Table 6:

Table 6. MODIS QA flags for the MCD43 product as well as VIIRS QA flag listed in https://lpdaac.usgs.gov/products/vnp43ia2v001/ for the VNP43 product

Code	Meaning
0	Best quality, full inversion
1	Good quality, full inversion
2	Magnitude inversion (Num Obs >=7)
3	Magnitude inversion (Num Obs>=3 & <7)
4	Fill value

We assume that the mean and uncertainty for a given pixel for category QA code 0 can be estimated from samples over the N years of observations 20+ years for MODIS and 8+ years of VIIRS, with a minimum of four samples and denote the mean $\bar{f}_{QA0,k}(i,j)$ and variance $\sigma^2_{QA0}(f_k,i,j)$ for some location (i,j) for parameter f_k. Of course this includes not only the uncertainty in the parameter but also any real variation in the parameter over the time period. It is therefore liable to be an over-estimate of uncertainty.

We then estimate f_k for other QA codes from its mean value for pixel (i,j) over the N-year time period, where a minimum of three samples exist and calls this $\bar{f}_{QAc,k}(i,j)$ for QA codes $c = \{1,2,3\}$. The assumption here is that $\bar{f}_{QA0,k}(i,j)$ represents f_k at (i,j) and characterise the departure for $\bar{f}_{QAc,k}(i,j)$ relative to $\sigma^2_{QA0}(f_k,i,j)$:

$$\sigma^2_{QA0}(f_k,i,j) = \frac{1}{N}\sum\left[\frac{\bar{f}_{QAc,k}(i,j)-\bar{f}_{QA0,k}(i,j)}{\sigma^2_{QA0}(f_k,i,j)}\right]^2 \quad [18]$$

where N is the number of samples over (i,j) (with 4 or more samples of QA 0 and 3 or more samples of QA c). We define a set of weighting terms W_{10}, W_{20} and W_{30} for each category of QA relative to QA0 where:

$$W_{c0} = \frac{1}{\sigma^2_{QAc0,k}} \quad [19]$$

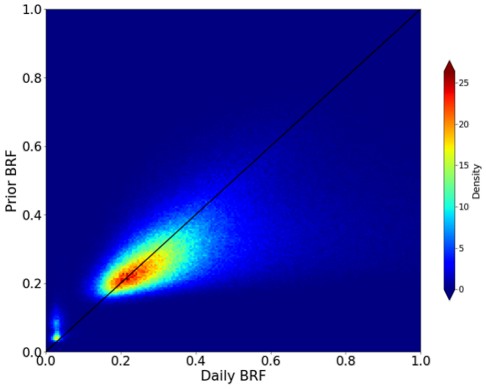
These weights can be used when calculating estimates of the mean value for each parameter for each pixel $\bar{f}_k(i,j)$ using data from all QA catagories:

$$\bar{f}_k(i,j) = \frac{1}{N_{(i,j)}}\sum_{c=0}^{c=3}\sum_{yQAc} W_{c0}\bar{f}_{QAc,k}(i,j) \quad [20]$$

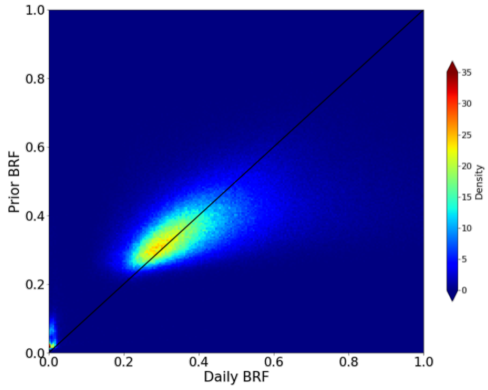
Defining $W_{00} = 1$. The summation over $yQAc$ is over all samples for pixel (i,j) that fall into the category QAc. The normalization term $N_{(i,j)}$ is:

$$N_{(i,j)} = \sum_{c=0}^{c=3} \sum_{yQAc} W_{c0} \quad [21]$$

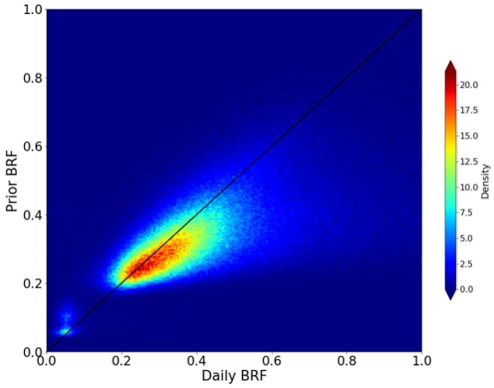
Figure 23 shows an example of comparison between daily and prior BRDF for MODIS tile h19v02 in year 2019 and day-of-year 33.



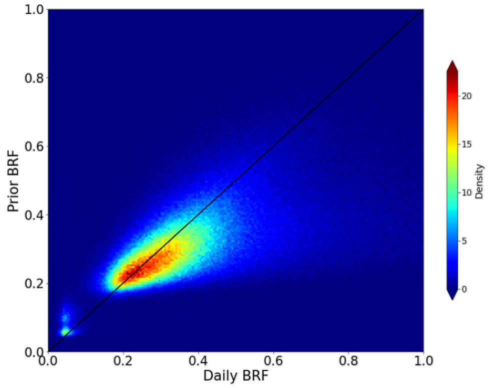
(a) Band-1 645nm



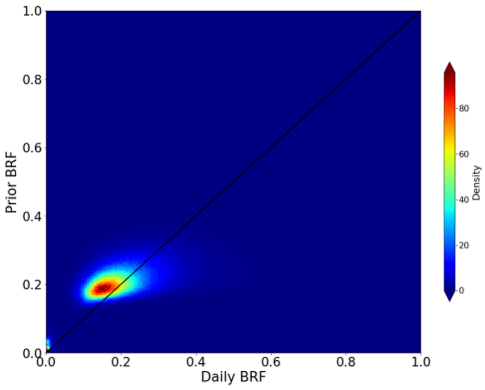
(b) Band-2 859nm



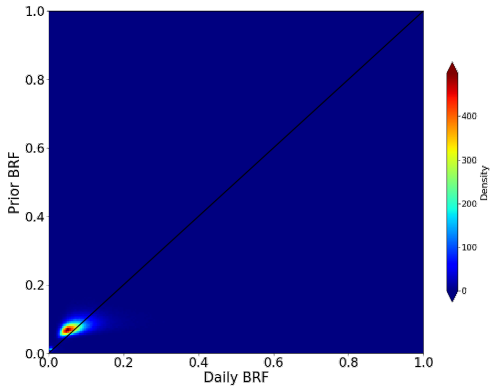
(C) Band-3 470nm



(d) Band-4 555nm



(e) Band-5 1240nm



(f) Band-1 1640nm

Figure 23. 2D scatterplots showing correlation between daily and prior BRDF for MODIS tile h19v02 in year 2019 and day-of-year 33.

2.9.1 Albedo calculation as a function of sun angle

The following polynomial has been found [39] to reproduce the Black-sky albedo, a_{bS}, well as a function of solar zenith angle, ϕ

$$a_{bS}(\phi,\lambda)=f_{iso}(\lambda)(g_{o_{iso}}+g_{1_{iso}}\phi^2+g_{2_{iso}}\phi^3)+f_{vol}(\lambda)(g_{o_{vol}}+g_{1_{vol}}\phi^2+g_{2_{vol}}\phi^3)+f_{geo}(\lambda)(g_{o_{geo}}+g_{1_{geo}}\phi^2+g_{2_{geo}}\phi^3) \quad [22]$$

where the g_{j_k} coefficients are listed in Table 7 below and the $f_k(\emptyset)$ are the BRDF model kernel weights or parameters. The integrated coefficients for the white-sky albedo are also provided.

g_{j_k} for kernel, k	k=isotropic	k=RossThick	k=Li-Sparse
g_{o_k}(term 1)	1.0	-0.007574	-1.284909
g_{o_k}(term)	0.0	-0.070987	-0.166314
g_{o_k}(term)	0.0	0.307588	0.041840
White-sky	1.0	0.189184	-1.377622

Table 7: *Coefficients for equation 22 to calculate albedo as a function of solar zenith angle*

3 Reference Documents

[1] Mahajan, S. and Fataniya, B., 2019. Cloud detection methodologies: Variants and development—A review. *Complex & Intelligent Systems*, pp.1-11.

[2] Irish, R.R., Barker, J.L., Goward, S.N. and Arvidson, T., 2006. Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm. *Photogrammetric engineering & remote sensing*, *72*(10), pp.1179-1188.

[3] Bley, S. and Deneke, H., 2013. A threshold-based cloud mask for the high-resolution visible channel of Meteosat Second Generation SEVIRI. *Atmospheric Measurement Techniques*, *6*(10), pp.2713-2723.

[4] Louis, J., Debaecker, V., Pflug, B., Main-Knorn, M., Bieniarz, J., Mueller-Wilm, U., Cadau, E. and Gascon, F., 2016. Sentinel-2 sen2cor: L2a processor for users. In *Proceedings Living Planet Symposium 2016* (pp. 1-8). Spacebooks Online.

[5] Zhu, Z. and Woodcock, C.E., 2012. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote sensing of environment*, *118*, pp.83-94.

[6] Zhu, Z., Wang, S. and Woodcock, C.E., 2015. Improvement and expansion of the Fmask algorithm: Cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images. *Remote Sensing of Environment*, *159*, pp.269-277.

[7] Foga, S., Scaramuzza, P.L., Guo, S., Zhu, Z., Dilley Jr, R.D., Beckmann, T., Schmidt, G.L., Dwyer, J.L., Hughes, M.J. and Laue, B., 2017. Cloud detection algorithm comparison

and validation for operational Landsat data products. *Remote sensing of environment*, *194*, pp.379-390.

[8] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.

[9] Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Van Esesn, B.C., Awwal, A.A.S. and Asari, V.K., 2018. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

[10] Ronneberger, O., Fischer, P. and Brox, T., 2015, October. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.

[11] Francis, A., Sidiropoulos, P. and Muller, J.P., 2019. CloudFCN: Accurate and robust cloud detection for satellite imagery with deep learning. *Remote Sensing*, *11*(19), p.2312.

[12] Wieland, M., Li, Y. and Martinis, S., 2019. Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network. *Remote Sensing of Environment*, *230*, p.111203.

[13] Jeppesen, J.H., Jacobsen, R.H., Inceoglu, F. and Toftegaard, T.S., 2019. A cloud detection algorithm for satellite imagery based on deep learning. *Remote sensing of environment*, *229*, pp.247-259.

[14] Zhang, Z., Iwasaki, A., Xu, G. and Song, J., 2019. Cloud detection on small satellites based on lightweight U-net and image compression. *Journal of Applied Remote Sensing*, *13*(2), p.026502.

[15] Mohajerani, S., Krammer, T.A. and Saeedi, P., 2018. Cloud detection algorithm for remote sensing images using fully convolutional neural networks. *arXiv preprint arXiv:1810.05782*.

[16] Mohajerani, S. and Saeedi, P., 2019, July. Cloud-Net: An end-to-end cloud detection algorithm for Landsat 8 imagery. In *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium* (pp. 1029-1032). IEEE.

[17] Khoshboresh-Masouleh, M. and Shah-Hosseini, R., 2020. A Deep Learning Method for Near-Real-Time Cloud and Cloud Shadow Segmentation from Gaofen-1 Images. *Computational Intelligence and Neuroscience*, *2020*.

[18] Molinier, M., Reunanen, N., Lämsä, A., Astola, H. and Räty, T., 2018, July. Deepcloud-A Fully Convolutionnal Neural Network for Cloud and Shadow Masking in Optical Satellite Images. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium* (pp. 2107-2110). IEEE.

[19] Li, Z., Shen, H., Cheng, Q., Liu, Y., You, S. and He, Z., 2019. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*, *150*, pp.197-212.

[20] Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F. and Adam, H., 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 801-818).

[21] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2014. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*.

[22] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), pp.834-848.

[23] Chen, L.C., Papandreou, G., Schroff, F. and Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

[24] Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, *111*(1), pp.98-136.

[25] Lazebnik, S., Schmid, C. and Ponce, J., 2006, June. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (Vol. 2, pp. 2169-2178). IEEE.

[26] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).

[27] Breiman, L., 2001. Random forests. *Machine learning*, *45*(1), pp.5-32.

[28] Rumelhart, D.E., Hinton, G.E. and Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, *323*(6088), pp.533-536.

[29] Winter, M.E., N-FINDR: An Algorithm for Fast Autonomous Spectral Endmember Determination in Hyperspectral Data. *Proceeding on SPIE Imaging Spectrometry*, 1999, pp.266-275.

[30] Heinz, D.; Chang, C.-I.; Althouse, M.L.G. Fully constrained least-squares based linear unmixing hyperspectral image classification. In *Proceedings of the IEEE 1999 International Geoscience and Remote Sensing Symposium*, Hamburg, Germany, 28 June–2 July 1999; pp. 1401–1403.

[31] Shuai, Y.; Masek, J.G.; Gao, F.; Schaaf, C.B. An algorithm for the retrieval of 30-m snow-free albedo from Landsat surface reflectance and MODIS BRDF. *Remote Sens. Environ.* 2011, 115, 2204–2216.

[32] Strahler, A.H.; Muller, J.; Lucht, W.; Schaaf, C.; Tsang, T.; Gao, F.; Li, X.; Lewis, P.; Barnsley, M.J. MODIS BRDF/Albedo product: *Algorithm theoretical basis document version* 5.0. MODIS Doc. 1999, 23, 42–47

[33] Feng et al. (2019) A Sensor Invariant Atmospheric Correction: Sentinel-2/MSI AND Landsat 8/OLI https://doi.org/10.31223/osf.io/ps957.

[34] Eskes, H. J., Antonakaki, T., Basart, S., Benedictow, A., Blechschmidt, 894 A.-M., Chabrillat, S., Christophe, Y., Clark, H., Cuevas, E., Hansen, 895 K. M., and Others (2018). Upgrade verification note for the CAMS 896 near-real time global atmospheric composition service: Evaluation of 897 the e-suite (experiment gkvv) for the period April 2016-October 898 2016. Technical report, Copernicus.

[35] Li. Z., et al. Preliminary assessment of 20-m surface albedo retrievals from Sentinel-2A surface reflectance and MODIS/VIIRS surface anisotropy measures. *Remote Sens. Environ.*, 2018, 217, 352-365.

[36] Bouvet, M.; Thome, K.; Berthelot, B.; Bialek, A.; Czapla-Myers, J.; Fox, N. P.; Goryl, P.; Henry, P.; Ma, L.; Marcq, S.; Meygret, A.; Wenny, B. N.; Woolliams, E. R. RadCalNet: A Radiometric Calibration Network for Earth Observing Imagers Operating in the Visible to Shortwave Infrared Spectral Range. Remote Sensing 2019, 11, 2401.

[37] Song, R., Muller, J.-P., Francis, A., Yin, F., Meygret, A., Farges, M. Validation of 10-m/20-m Sentinel-2 Albedo Products from the HR-AlbedoMap Retrievals. Presented at the first RadCalNet user workshop, 2021.

[38] Song, R., Muller, J.-P., Kharbouche, S., Woodgate, W. Intercomparison of Surface Albedo Retrievals from MISR, MODIS, CGLS Using Tower and Upscaled Tower Measurements. *Remote Sens*. 2019, 11, 644. https://doi.org/10.3390/rs11060644.

[39] Lucht, W., et al. (2000), An algorithm for the retrieval of albedo from space using semiempirical BRDF models, *IEEE Trans. Geosci. Remote Sensing*, *38*, 977-998