Training-free thick cloud removal for Sentinel-2 imagery using value propagation interpolation

Laurens Arp[a], Holger Hoos[b,a], Peter van Bodegom[c], Alistair Francis[d], James Wheeler[d], Dean van Laar[a], Mitra Baratchi[a]

[a]*Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands*
[b]*Chair for Artificial Intelligence Methodology, RTWH Aachen University, Germany*
[c]*Institute of Environmental Sciences, Leiden University, The Netherlands*
[d]*Φ-lab, European Space Research Institute, European Space Agency, Italy*

Abstract

Remote sensing imagery has an ever-increasing impact on important downstream applications, such as vegetation monitoring and climate change modelling. Clouds obscuring parts of the images create a substantial bottleneck in most machine learning tasks that use remote sensing data, and being robust to this issue is an important technical challenge. In many cases, cloudy images cannot be used in a machine learning pipeline, leading to either the removal of the images altogether, or to using suboptimal solutions reliant on recent cloud-free imagery or the availability of pre-trained models for the exact use case. In this work, we propose VPint2, a cloud removal method built upon the VPint algorithm, an easy-to-apply data-driven spatial interpolation method requiring no prior training, to address the problem of cloud removal. This method leverages previously sensed cloud-free images to represent the spatial structure of a region, which is then used to propagate up-to-date information from non-cloudy pixels to cloudy ones. We also created a benchmark dataset called SEN2-MSI-T, composed of 20 scenes with 5 full-sized images each, belonging to five common land cover classes. We used this dataset to evaluate our method against three alternatives: mosaicking, an AutoML-based regression method, and the nearest similar pixel interpolator. Additionally, we compared against two previously published neural network-based methods on SEN2-MSI-T, and evaluate our method on a subset of the popular SEN12MS-CR-TS benchmark dataset. The methods are compared using several performance metrics, including the structural similarity index, mean absolute error, and error rates on a downstream NDVI derivation task. Our experimental results show that VPint2 performed significantly better than competing methods over 20 experimental conditions, improving performance by 2.4% to 34.3% depending on the condition. We also found that the performance of VPint2 only decreases marginally as the temporal distance of its reference image increases, and that, unlike typical interpolation methods, the performance of VPint2 remains strong for larger percentages of cloud cover. Our findings furthermore support a cloud removal evaluation approach founded on the transfer of cloud masks over the use of cloud-free previous acquisitions as ground truth.

Keywords: Cloud removal, spatial interpolation, multispectral data, training-free, artificial intelligence

1. Introduction

Remote sensing data, such as the data obtained constantly from Earth observation satellites, is of tremendous importance in monitoring the health of our planet. However, when working with remote sensing data, data processing pipelines that could otherwise produce excellent results are often challenged by clouds obscuring parts of a satellite image. In some cases, these cloudy images are omitted entirely, even to the point of on-board hardware and software solutions being developed for satellites to avoid sending cloudy data back to Earth [1, 2]. Alternatively, cloud-free images are produced by a combination of cloud masking and mosaicking cloud-free pixels from a previous image onto the cloudy pixels of a target image [3, 4]. Although such an approach allows an application to accept the input image, pixel values from dynamic processes get outdated relatively quickly, and finding recent cloud-free images can be challenging. This can be a problem for tasks such as vegetation monitoring or the mapping of extreme events (e.g., floods or fires). Therefore, accurate and up-to-date estimations for cloudy regions would be much preferred, particularly in such dynamic environments.

Creating these cloud-free estimations for remote sensing images can be a challenging task. Environmental factors like the sun's azimuth and zenith angles, atmospheric conditions, vegetation, and the landscape change over time. A cloud removal algorithm would need to account for these

*Corresponding author
Email address: l.r.arp@liacs.leidenuniv.nl (Laurens Arp)

changes, which is difficult, considering the variability of the temporal distance to the last known cloud-free image. Similarly, the environment may have evolved in unpredictable ways, such as by extreme events or human activity. In recent years, deep learning-based cloud removal methods have shown strong performance compared to traditional methods. However, for downstream users, pretrained (deep) neural networks may require very precise combinations of input conditions such as the sensor, resolution and preprocessing, which may not be feasible for their use case, while developing new models for specific use cases is generally not trivial because new architectures, appropriate training configurations and (potentially large-scale) training datasets would be required.

In this article, we present a method to address these challenges by employing a technique that propagates the information in cloud-free pixels of the same image rather than using old pixel values. Specifically, we present a new cloud removal algorithm building upon our previously proposed spatial interpolation algorithm value propagation interpolation (VPint) [5]. In short, VPint iteratively calls a recursive update rule that propagates known values throughout a grid, applying spatial weights derived from a feature dataset. We have extended the algorithm to be suitable for the reconstruction of multispectral imagery. Our method uses the previously sensed imagery from the same time series as a feature dataset to inform the interpolation algorithm on the *spatial structure* of the underlying region of interest. It uses this structure to interpolate the reflectance values from the *current*, up-to-date image. In doing so, the current environmental conditions will also be propagated, rather than attempting to estimate these *a priori* to correct for them.

Applying VPint to optical remote sensing imagery introduces three main challenges: cloud removal being a remote sensing image processing rather than a general interpolation problem, temporal heterogeneity and exploding values: firstly, the original VPint algorithm leveraged machine learning models to predict the intensity of spatial autocorrelation between two spatially neighbouring points. However, in cloud removal, the intricate textures, transitions and objects in the image must be reproduced exactly, requiring a more precise representation of the spatial structure. Secondly, different sets of pixels may change in different ways between the feature dataset and target image, such as a (dynamic) crop field next to a (static) road. The relationship between these objects will change over time, introducing inaccuracies in the reconstructed target image. We refer to this problem as *temporal heterogeneity*. Thirdly, faulty pixels, solar glint or transmission errors can introduce erroneous values into the feature data, which can result in unrealistically large pixel value predictions that get propagated throughout the reconstructed image. We refer to this problem as *exploding values*. To address these issues, we propose VPint2, which incorporates a new method for computing spatial weights and includes extensions to the original VPint algorithm that alleviate problems caused by these phenomena: identity priority and elastic band resistance. Although VPint2 is aimed at improving VPint's applicability to optical remote sensing data, it is likely that it will similarly enjoy improved performance in other application areas where similar challenges arise, particularly in image processing. Through our experiments, we aimed to gauge the effectiveness of VPint2 as a cloud removal algorithm compared to existing methods, and to investigate under which conditions its performance is particularly strong.

Our contributions presented in this article are as follows:

- We propose a novel cloud removal method, leveraging the spatial structure from previously sensed imagery to propagate the non-cloudy values of the up-to-date cloudy image. Our method does not require a training phase and can be easily applied to any type of land surface data, requiring no additional data compared to pixel substitution approaches.

- We extended the spatial interpolation algorithm VPint to create VPint2, which modifies the algorithm to be applicable to remote sensing image processing problems, and features two enhancements we dub *identity priority* and *elastic band resistance*, improving its performance and its applicability to remote sensing datasets. We include an auto-adaptation mechanism to allow VPint2 to adapt its configuration to specific patches and bands.

- We created a benchmark dataset of 20 matched (target–cloud mask–temporal features) sets of top-of-canopy Sentinel-2 imagery, called SEN2-MSI-T. Unlike existing benchmarks, the true images are available as ground truth, as the clouds are derived from a different image, and the features are available at various specific time intervals. This allows for a better evaluation of methods, and the results from our experiments show that typical evaluation approaches using a recent cloud-free acquisition as ground truth can be problematic.

- We tested our method on SEN2-MSI-T and the existing SEN12MS-CR-TS benchmark dataset against mosaicking (temporal replacement), automated-machine-learning-based regression, similar pixel interpolation and neural-network-based approaches. Our experiments demonstrate that our method performs better than competing methods in all 20 conditions we tested in our main experiments, and in 17 out of 20 conditions in our experiments for Level 1C data.

The remainder of this article is organised as follows. In Section 2, we formalise cloud removal as a general interpolation problem; Section 3 covers the related work; Section 4 details our proposed method; Section 5 explains our experimental setup, and the results of our experiments are presented in Section 6. Finally, in Section 7, we draw some

general conclusions and briefly discuss avenues for future work.

2. Problem statement

Cloud removal can be formalised as a general spatial interpolation problem. Let image $\mathbf{O}$ denote the matrix representation of the original input image with clouds to be removed. This image consists of pixels $o_{ij} \in \mathbf{O}$, where i the row index, and j is the column index of the pixel, corresponding to the spatial position (i, j). Since all spectral bands in optical images are affected by thick clouds, cloud removal methods will typically need to be applied independently to all bands (we therefore introduce no additional band notation). Next, we use $\mathbf{T}$ to denote the matrix of the corresponding true (ground truth) cloud-free image, consisting of pixels $t_{ij} \in \mathbf{T}$, which would be unknown in practice, and $\mathbf{F}$, consisting of pixels $f_{ij} \in \mathbf{F}$, as the matrix of the cloud-free reference (feature) image obtained at some point in time prior to $\mathbf{O}$. Finally, we use $\mathcal{C}$ to denote the set of cloudy target image pixels o_{ij} for $\mathbf{O}$ (cloud mask). Our aim is to find a model $\mathcal{M}(\mathbf{O}, \mathbf{F}, \mathcal{C})$, taking the target image, feature image and cloud mask as input, and generating a predicted cloud-free image $\hat{\mathbf{T}}$ with pixels $\hat{t}_{ij} \in \hat{\mathbf{T}}$ resembling $\mathbf{T}$ as closely as possible. The problem thus becomes to find:

$$\mathcal{M}^* \in \underset{\mathcal{M}}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{T}, \mathcal{M}(\mathbf{O}, \mathbf{F}, \mathcal{C})) \tag{1}$$

Here $\mathcal{L}$ is the loss function of interest (for example, mean absolute error).

3. Related work

Given its importance to downstream remote sensing tasks, cloud removal in optical satellite data is of significant interest in the research community. Generally speaking, cloud removal must be guided by some type of information complementarity, which may be *spatial*, *multi-modal*, *temporal*, or a mixture of these. In addition to this, there are also cloud removal methods operating on the spectral domain to remove thin clouds (which, due to partial transparency, retain some surface information) [6, 7, 8, 9, 10, 11, 12, 13]. However, since we aim to remove all types of clouds along with their shadows, we do not consider this type of method further in this section.

Spatial methods rely on patterns within the cloud-free regions of an image to reconstruct cloudy regions. Much of the work we will refer to in this section contains a spatial component, for example, through the use of convolutional neural networks (CNNs) or the selection of suitable nearby pixels. However, most of these methods will also exploit other types of information complementarity. In contrast, general-purpose spatial interpolation techniques can also be used for single-image cloud removal by considering cloudy pixels (and cloud shadows) as missing data. This approach has been explored for other types of missing data in remote sensing imagery (sensor faults) by Zhang et al. [14]. However, many interpolation methods suffer from poor scalability, and Shen et al. [15] found that interpolation approaches are primarily effective at filling small gaps, such as the Landsat ETM+SLC-off dataset [16]. In the case of cloud removal, clouds can cover relatively large parts of an image which, combined with the high resolution of the imagery, results in large gaps, for which interpolation methods have not been popular nor especially successful so far.

Multi-modal methods exploit the information complementarity between different sensors, notably synthetic aperture radar (SAR), which penetrates cloud cover, to reconstruct cloud-free images. One of the most prominent examples of this type of cloud removal is DSen2-CR by Meraner et al. [17], which is based on convolutional neural networks (CNNs) and leverages SAR data. Xu et al. [18] proposed global-local fusion approaches to minimise performance degradation due to speckle noise and the domain gap between optical- and SAR data. Han et al. proposed a transformer-based approach for SAR-optical data fusion-based cloud removal [19], and Liu et al. proposed an attention-based network fusing Sentinel-2 and Sentinel-3 data sources [20]. General adversarial network (GAN) methods performing cloud removal using SAR-optical data fusion include the work by Xu et al. [21] and Darbaghshahi et al. [22], whereas Jing et al. proposed a method leveraging denoising diffusion [23]. Fusing optical data with other data sources, particularly SAR data, can be a challenging problem due to temporal shift, the noisy nature of SAR measurements, incomplete or non-overlapping spatial coverage at desired time steps, and the complex preprocessing pipelines SAR data typically require.

Multi-temporal methods exploit the temporal information complementarity of satellite imagery to gap-fill missing data. Here we differentiate between multi-temporal methods, which may exploit temporal information in a variety of ways, and the specific case of *time-series modeling* methods, which operate on a consistent time-series of images. A commonly used multi-temporal approach is mosaicking. Typically, a user would use cloud detection methods [24], such as s2cloudless [25] or SEnSeIv2 [26], to automatically detect cloudy pixels, fetch the most suitable non-cloudy pixel from past imagery (based, for example, on temporal distance), and mosaic these non-cloudy pixels onto a target image. We refer to this approach as *temporal replacement*, for which recent examples of practical use of this type of method include work on downstream tasks such as ecological monitoring [27] and (tree extent) mapping [28]. More sophisticated mosaicking approaches may account for changes in atmospheric conditions, solar azimuth and zenith angles, and other potentially confounding processes [3], or incorporate histogram matching [4, 29]. The accuracy of the reconstructed images will be greatly dependent on the availability of suitably recent cloud-free images. Pixel-wise re-

gression models can also be used to directly predict the pixel values of the target image using pixel values in the reference image as features. Due to the multicollinearity likely to be present in the different bands of the feature image, partial least squares (PLS) approaches tend to be preferred over ordinary least squares (OLS) models in remote sensing applications, as they compute inherently independent components [30, 31, 32]. Some approaches, such as the CHAIN and CROSS models proposed by Fischer et al. [33], use graphical and probabilistic models instead of grid-based spatial statistical models. Some methods, such as (M)NSPI [34, 35], WLR [36], STMRF [37], CLMP [38] and STWR [39], use temporal information to predict reconstructed pixel values from a local spatial neighbourhood of matching pixels.

Time-series modeling methods are a specific case of multi-temporal methods in which the full time-series of (possibly cloudy) regular temporal acquisitions by satellites is exploited for information complementarity. Zhu et al. [40] proposed the use of three time-series models, at varying levels of complexity, to predict missing pixel values in time-series of Landsat images, wheras SSTC-CR by Zheng et al. [41] leverages tensor decomposition to model various relationships between the spatial, spectral and temporal domains of the time-series of satellite data. More recently, multiple neural network-based approaches have been proposed for multi-temporal cloud removal, often incorporating multi-modal data as an additional component. UnCRtain-TS by Ebel et al. [42] combines a multi-temporal approach with SAR-optical data fusion, and supports uncertainty estimates on its predictions. Zhang et al. [43] applied a CNN on fused temporal features, Zhang et al. [44] applied a CNN on decomposed tensors, Zhao et al. [45] leveraged diffusion models for cloud removal on time-series data, and Zou et al. proposed a fast diffusion approach using SAR data [46]. Stucker et al. [47] used convolutional layers to encode and decode spatial information, while performing temporal attention on the resulting latent representations of individual time steps. Multi-temporal approaches are typically less applicable to scenarios where only one or a few target images are relevant to the user, or when there are large or inconsistent temporal gaps in data.

In the following, we address weaknesses in the existing work by proposing a novel cloud removal method called VPint2, which extends the value propagation interpolation (VPint) [5] algorithm. Interpolation methods, due to the weaknesses of existing methods in gap-filling large clusters of missing data, have not been explored much for the purpose of cloud removal. Our proposed method, overcoming the weaknesses of existing interpolation methods, therefore offers a novel branch of research in cloud removal as a multi-temporal (single reference image) interpolation method.

4. Methods: Value propagation-based interpolation (VPint)

The VPint algorithm[1] was inspired by Markov reward processes [48], particularly their application in reinforcement learning settings to estimate state values using immediate rewards and discounted future rewards. When applied to cloud removal, given a cloudy target image and a previously sensed non-cloudy image as features, it returns a cloud-free reconstructed target image. The process is applied independently for all spectral bands, thus being robust to the typically diverse spatial patterns in different bands of remote sensing imagery, and offering a considerable potential for parallelisation for imagery with a large number of spectral bands. The general workflow for applying VPint2 to cloud removal is shown in Figure 1.

4.1. Value propagation interpolation algorithm

VPint [5] is a spatial interpolation algorithm that recursively propagates known values through unknown values, multiplied at every recursive step by spatial weights representing spatial autocorrelation between pairs of pixels. A spatial weight < 1 would represent a decrease in value for the original pixel compared to a neighbouring pixel, and a weight > 1 would represent an increase thereof. In the case of cloud removal, cloudy pixels (and cloud shadows) would be unknown values, whereas non-cloudy pixels would be known values.

The spatial weights can be static, in which case no feature image is needed, and values regress to an initialisation value (typically the mean) with distance. However, the main benefit of using VPint for cloud removal stems from its support for location-specific data-driven weights. In this case, features derived from a reference image can inform the choice of weights for the VPint algorithm, by containing information on the spatial structure of the underlying scene. In the original VPint algorithm, these weights could be predicted from other, related variables using machine learning models. This allowed the algorithm to make relatively smooth predictions, based mainly on spatial autocorrelation. In optical satellite data, images have sharp edges, textures and other challenging, abrupt changes, calling for a different approach to computing spatial weights. Therefore, we propose a modification to the core of VPint to directly compute exact spatial weights at runtime, without the need of an explicit optimisation or training procedure, making it a training-free method. We will refer to this updated algorithm as VPint2.

In optical satellite data, if two images are taken of the same area at different times, the spatial structure of the land surface of two co-registered images sensed at different times would remain (mostly) static. For example, a

[1]The code for VPint2, which supercedes the original VPint algorithm, can be found on GitHub at `https://github.com/ADA-research/VPint2`

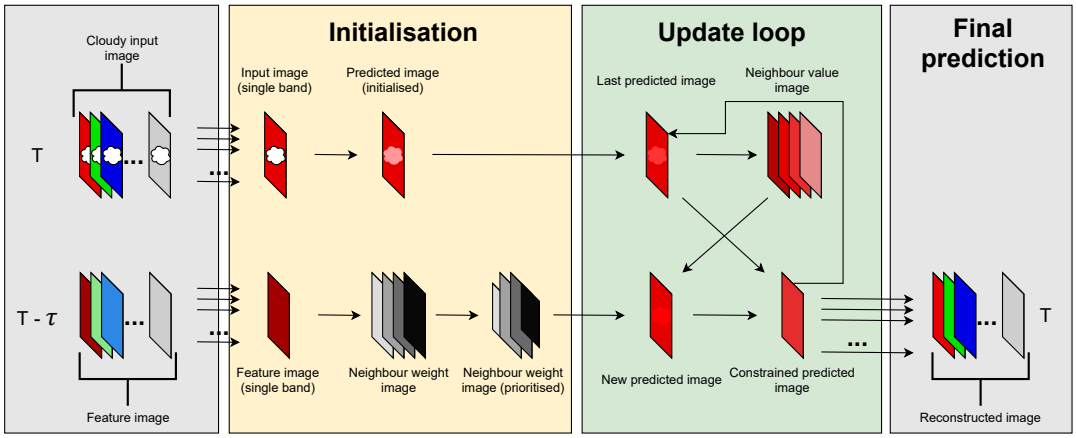


Figure 1: General workflow for applying VPint2 to cloud removal problems. First, the input- and feature images are split up into individual bands (for which we show one example in the red band). Next, for every band, the predicted image is initialised to fill cloudy pixels with the mean value of the image (other initialisations are possible), and the spatial weights for the neighbours of every pixel are computed and stored in a 4-channel image, where every channel represents a direction. In this image, for every pixel, the channels contain weights by which a neighbouring value would need to get multiplied to form the pixel's value in the feature image. If identity priority is used, these weights are then refined into a re-weighted neighbour weight image with lower impact for more extreme weights. After this, in the update loop, a new image is computed iteratively by multiplying a neighbour value image (with predicted image values for neighbouring pixels in channels corresponding to those of the weight image) with the weight image. If using elastic band resistance, the growth of values in the resulting new predicted image is constrained by comparing the previous predicted image to that of the current iteration. After auto-termination (or a specified maximum number of iterations has been reached), the most recent (constrained) predicted image becomes the predicted image for this band. All bands are then combined again to form the cloud-removed output of VPint2.

residential suburb may change in hue as time passes and seasons change (dynamic values), but whether it is shiny in the sun or snowed over in winter, the similarity between the pixels within houses, trees and gardens the field will remain relatively high (static spatial structure). This means that the weights within these structures should remain close to 1 (a weight of 1 between a pair of pixels signifies that they are identical). Similarly, the weights on the border can be expected to be further removed from 1, as the neighbouring pixels on the border will be more dissimilar from one another, requiring a transition for the values being propagated.

This intuition gives rise to the notion of *objects* in an image. Pairs of pixels belonging to the same object (*same-object* pairs) will have weights close to 1, and pairs of pixels not belonging to the same object (*different-object* pairs) will have weights further removed from 1. These objects need not be explicitly defined (i.e., no object detection algorithms are necessary). Instead, they are contained in the spatial weights derived from the reference image.

This consistency of the spatial structure over time is the property we exploit with VPint2, by assuming temporally static spatial relationships for temporally dynamic values, and feeding the reference image as a feature set to the algorithm. The manner of deriving weights from a feature image can vary, and has a high impact on the behaviour of the method. The predictions from a machine learning model, as used in the original VPint algorithm, could not easily model the strong and abrupt changes in remote sensing image processing tasks. However, when using very precise weights with sufficient variability to be applicable to images, mistakes and errors can also have a larger impact on performance. Therefore, one of the challenges in applying VPint to remote sensing (and possibly general image processing) tasks is to use an approach for deriving spatial weights that is both exact and reliable, while mitigating the risks of large errors that exact, non-smoothed weights entail.

To address this problem, we can leverage the property of satellite data automatically revisiting the same area at specific time intervals. Although the latest cloud-free reference image could be months in the past, especially given the temporal autocorrelation of cloudy and rainy weather, we can exploit these reference images in VPint2 by extracting the spatial structure of a location, using the past reference image as a feature image to compute highly accurate spatial weights. A simplified illustration of how this spatial structure is used can be found in Figure 2.

Concretely, we instantiate the predicted image $\hat{\mathbf{T}}$ with pixels $\hat{t}_{ij}$ by copying non-cloudy pixels $o_{ij} \notin \mathcal{C}$ from $\mathbf{O}$, and initialise $\hat{t}_{ij}$ for all cloudy pixels $o_{ij} \in \mathcal{C}$ as the mean value of $\mathbf{O}$ (other initialisation approaches are possible). Let $N(i,j)$ denote the set of neighbouring positions (i',j') for a given position (i,j). With τ denoting the current iteration, we iterate the following update rule:

$$\hat{t}_{ij}^{\tau+1} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ \frac{1}{|N(i,j)|} \cdot \sum\limits_{(i',j') \in N(i,j)} \gamma_{ij}^{i'j'} \cdot \hat{t}_{i'j'}^{\tau}, & \text{otherwise} \end{cases} \quad (2)$$

The update rule of Equation 2 contains a number of important elements. First, all non-cloudy pixels (that is, $o_{ij} \notin \mathcal{C}$) merely take on the static value of the input image. For cloudy pixels, at every iteration, a local prediction is computed, using the weighted average of predicted values

Figure 2: A basic, one-dimensional, unidirectional example illustrating the process of VPint2. The middle two cells in the target image are unknown, whereas the reference image is fully known. Although the values themselves lie in a different range (multiplied by 10 in this simplified example), the relationships between neighbouring cells are the same in this example. By computing weights from the reference image, we can interpolate the unknown values in the target image. In reality, interactions would bi-directional, and the problem would have two spatial dimensions.

of neighbouring unknown pixels (or static known values where available). Here $|N(i,j)|$ denotes the number of neighbours to pixel $\hat{t}_{ij}$ (in practice 4 for all pixels apart from the image edges), and $\gamma_{ij}^{i'j'}$ is the spatial weight between the pixels at positions (i',j') and (i,j). This weight is computed from the corresponding pixels in feature image $\mathbf{F}$ as:

$$\gamma_{ij}^{i'j'} = \frac{f_{ij}}{f_{i'j'}} \tag{3}$$

Thus the value of every cloudy pixel in $\hat{\mathbf{T}}$ is determined by the values of its local neighbours, which, if cloudy, are themselves determined by their neighbours. At every iteration and recursive step, the neighbouring values are multiplied by a spatial weight derived from the feature image, allowing the algorithm to incorporate complex structures, textures and variability within subsets of the image.

By iterating Equation 2, pixel values are updated repeatedly, anchored by non-cloudy target values that are propagated following the spatial structure given by $\mathbf{F}$ (through γ), until an equilibrium configuration is reached.

4.2. VPint2 properties

VPint2 has a number of desirable properties that existing cloud removal methods do not yet offer. First, VPint2 estimates the current state of the measured quantity, as opposed to methods merely copying previous information (which was already known). Second, it offers the advantage of not needing any training, thus avoiding the problems of methods that need to either attempt to train one general model applicable to all cases, extrapolate from a specialised model that does not generalise well, or train on a prohibitively small training set of non-cloudy pixels from the same image. Third, the results and inner workings of VPint2 can be understood by analysing the reference image in combination with the update rule of Equation 2.

On the other hand, as an interpolation method, VPint2 requires at least one non-cloudy pixel in the input data, otherwise it would simply reconstruct the feature image. This might make it less suitable for cloud removal in smaller patches, where the probability of all pixels being cloudy is higher.

In terms of computational cost and efficiency, the running time of VPint2 will depend on its implementation. However, the implementation-independent efficiency can be approximated by drawing a parallel with neural networks, which are often described based on the number of parameters in the network. Since both neural networks and VPint2 are based on matrix multiplication, the computational effort mainly stems from the amount of multiplications and matrix elements that must be multiplied. In the case of VPint2, the time complexity to run on a single band is $\mathcal{O}(e \cdot h \cdot w \cdot 4)$, where e is the number of iterations or epochs for which VPint2 will be run (typically around 20), h is the height of the input image, and w is the width of the image. For example, if we run VPint2 for 100 iterations on a 100×100 image, resulting in 4 000 000 multiplications, the entire pipeline of VPint2 would be the equivalent of running *inference only* on a 4 million parameter neural network, while not requiring a prior training step. In practice, VPint2 will generally perform more frequent, smaller matrix operations compared to a typical neural network with a similar number of parameters, resulting in a slower running time. On the other hand, the multispectral nature of optical satellite data allows for great opportunities in parallelisation over bands, since VPint2 considers these bands independently. Therefore, we have extended the original VPint algorithm with a multiprocessing setup for Earth observation imagery, resulting in a substantial speedup of about 60% to 70% compared to the original algorithm (for empirical results, see Figure 7c). In the future, combining the parallelisation with GPU-accelerated matrix computations may speed up the algorithm even further.

4.3. Further enhancing VPint2 for remote sensing data

Applying VPint2 to remote sensing imagery comes with particular challenges, some of which may also be encountered in general image processing problems. In particular, VPint2 will perform worse when i) objects in the images changed over time between the feature and target images in different ways (e.g., one stays constant while another changes hue), which we will refer to as *temporal heterogeneity*, and ii) sensor faults or other inaccuracies are present in the feature set, resulting in extremely large weights (and extremely large values that get propagated further), which we refer to as *exploding values*.

To illustrate the problem of temporal heterogeneity, recall the concept of *objects* introduced in Section 4.1. In the case of removing clouds from optical imagery, the *within-object* relationships (weights close to 1) will typically be easy to exploit, whereas *between-object* relationships will be less reliable. For example, a road next to a forest will remain mostly static throughout the seasons, while the forest may be shedding and gaining leaves over time. This means

that the spatial weights between the road and the forest computed from the reference image (for example, one from a summer, when the forest was full of green leaves) will no longer apply to the new between-object relationships (for example, one from an autumn, when the leaves may be gone, or yellow and brown). At the same time, the internal homogeneity of both objects will generally be mostly intact. The original VPint2 algorithm, however, relies equally on within-object and between-object weights, and can therefore suffer from artefacts and other inaccuracies caused by temporal heterogeneity. Visually, such artefacts would look like a fading gradient of an incorrect colour that is strong at the borders of objects and gradually fades into the colour hue of the rest of the object.

We consider an image reconstruction to suffer from exploding values when the VPint2 algorithm is diverging from, rather than converging to, a stable solution. This problem can arise in rare cases, because image data, and remote sensing imagery in particular, can suffer from inconsistencies, faulty pixels and other quality issues. If this occurs in the target image, these pixels can be treated as 'missing' and interpolated along with cloudy pixels, as long as the issues are identified in advance. However, some possible causes of quality issues in the data, such as solar glint or transmission errors, are not always easy to detect automatically. Moreover, if the issue exists in the feature image, it cannot simply be interpolated even if detected accurately. Because the weights derived from a faulty pixel can introduce an unrealistically large weight into the system, values multiplied by this weight can then be amplified too much by other weights and propagated along to other pixels as well. Similarly, the location of the border between objects can move over time. If the border between objects lies within the cloudy region in the feature image, but outside the cloudy region in the target image, it would be wrongly applied, despite the transition having already occurred in the target image (the opposite case of not being applied at all is also possible). In both of these cases, a disruption in the balance of the system of weights would cause unreasonably large values to be estimated, passed on to their neighbours, and grow at an even faster rate in the next iteration, resulting in 'exploding values' in that area.

Addressing these challenges, we propose two technical enhancements to the VPint2 method, thereby boosting its general performance and its applicability to remote sensing imagery.

4.3.1. Identity priority

Given the problems caused by temporal heterogeneity, our first enhancement of VPint2 aims to exploit reliable within-object relationships, while minimising the impact of less reliable between-object relationships. Since between-object relationships are more likely to suffer from temporal heterogeneity than same-object relationships (though both are possible), prioritising the information from neighbours belonging to the same object can alleviate the impact of this problem. To this end, we extended the VPint2 algorithm by incorporating *identity priority*. Recall Equation 2, where $\hat{t}_{ij}^{\tau+1}$ for unknown values was updated to the weighted average of neighbouring values. In effect, this update rule computes four independent predictions $\gamma_{ij}^{i'j'} \cdot \hat{t}_{i'j'}^{\tau}$ for every neighbouring pixel at positions (i', j'), and VPint2 later uses the mean thereof as its prediction. We realise identity priority by using a weighted mean instead of an average prediction. Specifically, we assign a priority weight $\lambda_{ij}^{i'j'}$ (separate from the *spatial* weight $\gamma_{ij}^{i'j'}$) to an individual prediction based on the spatial weight's distance to 1 (which would signify a same-object relationship). By computing $\lambda_{ij}^{i'j'}$ to be equal to $\gamma_{ij}^{i'j'}$ for spatial weights ≤ 1, but dividing 1 by $\gamma_{ij}^{i'j'}$ for spatial weights > 1, this distance is equal in both directions. As a result, we prevent bias towards over- or underestimations, as a spatial weight indicating a halved value (0.5) would have the same priority weight as a spatial weight indicating a doubled value (2). Moreover, in some cases it may be beneficial to increase or decrease the degree to which weights close to 1 are favoured. To control this intensity, we introduce a new parameter β, which determines the strength of the identity priority procedure. Thus, priority weights are computed as:

$$\lambda_{ij}^{i'j'} = \begin{cases} \beta \cdot \gamma_{ij}^{i'j'}, & \text{if } \gamma_{ij}^{i'j'} \leq 1 \\ \beta \cdot \frac{1}{\gamma_{ij}^{i'j'}}, & \text{otherwise,} \end{cases} \tag{4}$$

and if we denote the sum of priority weights $\sum_{(i',j') \in N(i,j)} \beta \cdot \lambda_{ij}^{i'j'}$ as Λ_{ij}, the modified version of Equation 2 becomes:

$$\hat{t}_{ij}^{\tau+1} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ \frac{1}{\Lambda_{ij}} \cdot \sum_{(i',j') \in N(i,j)} \lambda_{ij}^{i'j'} \cdot \gamma_{ij}^{i'j'} \cdot \hat{t}_{i'j'}^{\tau}, & \text{otherwise} \end{cases} \tag{5}$$

The choice for the identity priority intensity parameter β can be highly relevant to the error rates of the algorithm. If it is set too high, error rates tend to increase on images (or regions thereof) where the between-object relationships did not change much in the time between the target and reference images, even if the results still look plausible to the human eye. Conversely, if β is set too low, some images may suffer from higher error rates, and visual artefacts may appear due to the temporal heterogeneity. We opted to automatically adapt β based on the performance of the algorithm on a validation set derived from the available cloud-free pixels, using the procedure described in Section 4.3.3. Regardless of the strength of β, identity priority cannot fully prevent artefacts from occurring at the borders of objects in specific cases, namely when two different objects in the target image were part of the same object in the feature image (thus having weights close to 1).

4.3.2. Elastic band resistance

The problems caused by exploding values can be substantial, particularly due to the value propagation-based nature of VPint2 propagating these errors throughout the entire image. To some extent, the impact of such errors can be mitigated by clipping the possible values in an image, but in this case, the exploded values would still greatly hamper performance by filling parts of the image with the user-defined maximal value. Moreover, such a hard threshold would leave little room for unlikely, but physically meaningful, high values. Therefore, we propose to combine a conservatively used hard threshold (that should, ideally, never be met) with a soft additional enhancement to VPint's update rule, which we refer to as *elastic band resistance*. This enhancement is aimed at improving performance on specific pixels, as well as preventing unreasonable predictions from propagating and possibly amplifying further, while still allowing exceptionally large increases in values where exceptional circumstances call for it. As an analogy, we can compare the increase of values at a certain pixel to stretching an elastic band. Up to a certain threshold, in this case the length of the elastic band, it can be moved freely. However, beyond this threshold, the further one stretches the band, the higher the resistance will be, and the more force needs to be applied to achieve even a tiny amount of additional length. This behaviour can be modelled using Hooke's law:

$$F = k \cdot \Delta L \tag{6}$$

Here, F is the force required to stretch an elastic band for an additional length ΔL. This is controlled by the resistance k, where high values for k require higher amounts of force for smaller stretching lengths.

We can adapt Equation 6 to our update function. If we define μ as the threshold beyond which we wish to apply resistance and $\Delta\hat{t}_{ij}^{\tau+1}$ as the amount of change between $\hat{t}_{ij}^{\tau}$ and $\hat{t}_{ij}^{\tau+1}$ after running Equation 2 or 5, we can update an old pixel value $\hat{t}_{ij}^{\tau}$ to its new value $\hat{t}_{ij}^{\tau+1}$ as:

$$\hat{t}_{ij}^{\tau+1} = \begin{cases} \hat{t}_{ij}^{\tau} + \Delta\hat{t}_{ij}^{\tau+1}, & \text{if } \hat{t}_{ij}^{\tau+1} < \mu \\ \hat{t}_{ij}^{\tau} + \Delta\hat{t}_{ij}^{\tau+1} - k \cdot \hat{t}_{ij}^{\tau}, & \text{otherwise} \end{cases} \tag{7}$$

Since we use $\hat{t}_{ij}^{\tau}$ instead of $\Delta\hat{t}_{ij}^{\tau+1}$ as our penalty term, Equation 7 deviates somewhat from Hooke's law as stated in Equation 6. However, this provides us with the desired property: the larger the absolute difference between the previously predicted value $\hat{t}_{ij}^{\tau}$ and μ, the stronger the resistance applied by $k \cdot \hat{t}_{ij}^{\tau+1}$ will be, even if the force remains constant.

By applying this penalty term to the VPint2 update rule, drastic increases in values caused by exploding values could be dampened to a great extent, while this dampening would be much weaker on lower, more reasonable values. As a result, this functionality can address the problem of exploding values, provided the parameters μ and k are set appropriately. Much like β, overly aggressive settings for these parameters would result in higher error rates, due to values being unable to increase as far as they should. Thus, for these parameters as well, proper configuration is key in the performance of VPint2 for cloud removal, which we achieved through auto-adaptation.

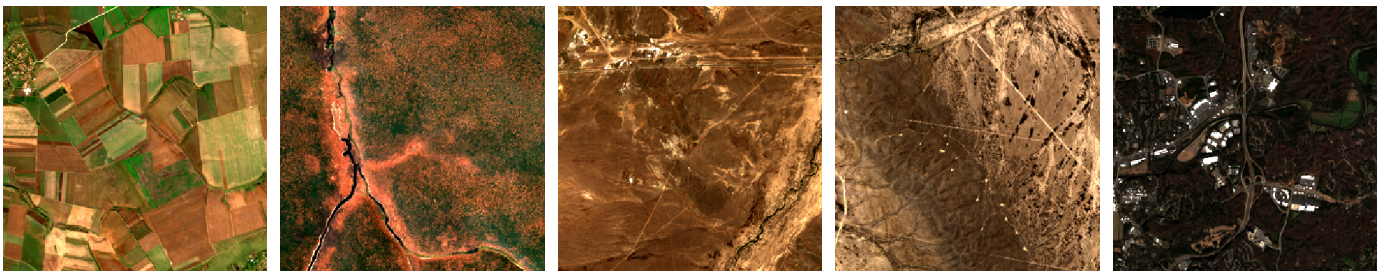
4.3.3. Auto-adaptation

Identity priority and elastic band resistance introduce new parameters, which we propose to set automatically using auto-adaptation. When using proper configurations for β, μ and k, these extensions successfully alleviate problems VPint2 would encounter when applied to remote sensing imagery. They may also be effective at alleviating similar problems in other applications that suffer from faulty pixels or changing spatial structure, such as video processing or pipelines reliant on noisy measurements. However, as explained in Section 4.3.1 and 4.3.2, inappropriate settings can have deleterious effects on the performance of VPint2. Moreover, the best performing parameter settings tend to vary greatly between scenes, patches within a scene, and even the spectral bands within the same image. As a result, a single configuration for a full image will not perform optimally, since the performance gains in one area may come at the expense of losses in another. On the other hand, manually selecting the appropriate parameter settings for all 12 bands of all 400 patches in a scene would not be feasible. Therefore, the automatic configuration of VPint2 is essential to its successful application.

We added a self-adaptive mechanism for automatically setting β, μ and k to appropriate values. This mechanism leverages the available data by sub-sampling known datapoints into a validation set, where pixels with the greatest mismatch between the target and feature images are prioritised. The adaptation algorithm can then search for appropriate parameter settings by sampling from the parameter space of possible configurations, and running VPint2 on the image with additional datapoints missing. This allows the algorithm to assign a validation loss to every parameter configuration that is sampled. The algorithm supports grid search and random search in its current implementation, but in principle, any black-box optimisation algorithm can be used. To ensure that performance will be *at least* on par with the original VPint algorithm, a configuration with no identity priority or elastic band resistance is always sampled first, although it is possible that the available validation pixels are not representative for some patches.

5. Experiments

This section explains the experimental setup used to evaluate the performance of VPint2 on cloud removal tasks. We first explain the questions that motivated our study and then cover the data we use in our experiments, the methods against which we compare, and our experimental setup.



(a) Cropland (b) Forest (c) Herbaceous (d) Shrubs (e) Urban/built up

Figure 3: Example target patches for the five land cover classes of our benchmark dataset. Figure 3a shows a patch from cropland in Hungary, Figure 3b shows a patch from a forested area in Angola (for which the target is from a dry period, but features for 6 months are green), Figure 3c shows a patch with herbaceous vegetation from Kazakhstan, Figure 3d shows a patch with shrubland from Mexico, and Figure 3e shows an urban patch from the US.

5.1. Questions addressed in our experiments

Our experiments were aimed at answering the following questions (which we later refer to as Q1, Q2, Q3 and Q4):

1. **Can identity priority and elastic band resistance improve VPint2's applicability to remote sensing imagery?** We lead with this question, as the answer to it will determine how VPint2 is used throughout the rest of our experiments.
2. **Can VPint2 achieve significantly better results than competing methods?** We quantitatively evaluate this across scenes of different land cover classes, taking advantage of the diversity in our dataset to allow different methods to perform well on images from different types of ecosystems and geography.
3. **How do the temporal distance between target and feature images, and the percentage of cloud cover in the input image, affect the relative performance of the methods?** In many cases it can be difficult to procure recent cloud-free images, and interpolation methods are typically better at gap-filling small amounts of missing data, making these meaningful variables to study. Moreover, since VPint2 requires a cloud-free reference image as features, it should still be able to perform well for higher temporal distances. We measure the performance against four temporal distances of 1 week, 1 month, 3 months, and 6 months, and per-patch performance against the percentage of cloudy pixels in the patches.
4. **How well does VPint2 perform in terms of running time compared to competing methods?** Running time can be an important factor in the practical use of a cloud removal method. We compare the average running time for different levels of cloud cover, as this is often a key factor in the computational efficiency of cloud removal methods.
5. **What is the overlap between the errors produced by different methods, and how could complementary strengths be exploited?** We visualise errors between methods, and explore the impact of ensembling strategies for improved performance.

5.2. Data

In our experiments, we focused on multi-spectral Sentinel-2 imagery (level 2A) as a use case for our method. The two Sentinel-2 satellites from the European Space Agency (ESA) measure reflectance values at 13 wavelength bands. The RGB bands (2,3,4) and band 8 have a spatial resolution of 10 m^2, bands 5,6,7,8A,11 and 12 have a spatial resolution of 20 m^2, and bands 1,9 and 10 have a resolution of 60 m^2. Band 10 is dropped in level 2A images, as it is mainly used for atmospheric correction, leaving 12 spectral bands to remove clouds from. We found existing benchmarks datasets to be scarce, and the few available benchmarks, such as SEN12MS by Schmitt et al. [49], its derivative SEN12MS-CR by Meraner et al. [17] and WHUS2-CRv by Li et al. [12], typically use previous cloud-free acquisitions as ground truth values and do not contain past imagery at various time intervals as feature data, making them inapplicable to our use case. Therefore, we created the SEN2-MSI-T benchmark dataset (referring to Sentinel-2, the optical multi-spectral instrument used by Sentinel-2, and the temporal aspect of the dataset) for our experiments, inspired by the existing datasets mentioned above.

SEN2-MSI-T. This dataset contains co-located optical imagery and a cloud mask for the five most common land cover classes in the Copernicus Global Land Cover [50] dataset: cropland, forest, herbaceous (vegetation), shrubs, and urban/built up area. For every type of land cover, we manually defined multiple search areas predominantly filled with the same land cover, and automatically searched for candidate scene locations meeting our requirements, from which we selected 4 per land cover class, resulting in a total of 20 scenes. Each of these scenes, which we strove to obtain from diverse geographical locations from Europe, Asia, Africa, the Americas and Australia, contains a cloud-free target image sensed at time s. It also contains a matching cloud mask, obtained from a cloudy image from a time as close as possible to s. It furthermore contains

four feature images at different (approximate) time intervals: $s-1$ week, $s-1$ month, $s-3$ months, and $s-6$ months. Thus, every scene consists of 6 different co-located large images in total. Each scene was partitioned into non-overlapping patches of 256×256 pixels, examples of which can be seen in Figure 3. Candidate solutions were identified, visualised and manually inspected using SentinelHub, and were downloaded as full level 2A data products using SentinelSat. At runtime, patches were loaded using windowed reading and resampled to a $10 \times 10m^2$ resolution. We provide the code to generate the dataset, along with download locations for the (compressed) raw data, in the code repository accompanying this article.

In many existing cloud removal datasets, such as SEN12MS-CR [17], SEN12MS-CR-TS [51] and WHUS2-CRv [11], model training supervision and performance evaluation is performed by matching cloudy target images with cloud-free acquisitions from one or two satellite revisits before the target (in the case of Sentinel-2, the revisit time is generally 5 days). The advantage of this approach is that real cloudy input data is used, whereas synthetic data experiments may have poor generalisability to real-world data, due to unrealistic cloud profiles, the common types of clouds in real data varying based on geographic properties, and the visual representation of the cloud itself. However, even with small temporal distances, the pixel values of a scene may have changed substantially, potentially reducing the reliability of this type of evaluation approach (our results in Section 6.3 will support this intuition empirically). Therefore, to allow for an accurate validation of our cloud removal approach, the clouds used in our experiments were masked from a separate image, which was then applied to a cloud-free image, as this allowed us to compute accurate, up-to-date performance measures. Our evaluation approach, therefore, offers a middle ground between accurate performance metric computation, and realistic cloud cover suitable for the specific geographical location of the scenes. A similar approach was recently successfully employed by U-TILISE [47], although unlike in our approach, missing (cloudy) pixels were denoted using the maximal pixel value, instead of using explicitly missing data points.

SEN12MS-CR-TS [51]. This dataset contains time-series of multiple ROIs with Sentinel-2 Level 1C imagery. The ROIs are split into 256×256 pixel patches, with 30 potentially cloudy images available for every patch. By performing experiments on this dataset we were able to compare our method against many state-of-the-art methods, such as STGAN [52], U-TAE [53] and UnCRtainTS [42], while also serving as a frame of reference to compare against future methods that are evaluated on this dataset. However, we note that our method was not intended to be used for time-series cloud removal, and could therefore only run on a subset of the test dataset, as determined by the following criteria. To evaluate our proposed method, we required i) a cloudy target image with at least one non-cloudy pixel, ii) a cloud-free reference image, and iii) a cloud-free ground-truth image for evaluation. We could simulate a dataset satisfying these criteria by computing cloud masks for every time step for the patches in SEN12MS-CR-TS, and selecting the patches for which we could identify a combination of temporally close ground truth and target images, with a cloud-free reference image available at some time step prior to the target.

When making use of VPint2 for cloud removal on real-world cloudy input data, such as SEN12MS-CR-TS, users should take care to incorporate a high quality cloud masking algorithm. In our experiments on SEN12MS-CR-TS, we used the SEnSeI-v2 cloud detection model [26] to generate cloud masks, whereas experiments on SEN2-MSI-T used the cloud probability band of the Sentinel-2 Level 2A data products. In general, recall should be prioritised over precision for cloud masks for VPint2. Although high recall may come at the cost of lower precision, needlessly interpolating a few cloud-free pixels will not have a large impact on the performance of the algorithm. On the other hand, wrongly accepting cloudy pixels as true values, and thus propagating cloudy pixel values throughout the image, could have a substantial negative impact. In a similar vein, buffering cloud masks is recommended to ensure full masking around the edges of clouds.

5.3. Competing and alternative methods

We compared the performance of VPint2 to that of several alternative state-of-the-art methods. We strove to include in our selection representative methods from all the categories listed in Section 3 (apart from interpolation methods due to scalability issues of competitive methods). Therefore, we compare the performance of VPint2 to that of temporal replacement, automated machine learning (AutoML)-based regression ensembling, and a deep neural network specifically designed for cloud removal. The approach most comparable to our proposed method is temporal replacement, since it also requires no training and only relies on a past cloud-free reference image. The AutoML regression and deep neural network methods have more requirements, due to their reliance on training (as well as model selection and hyperparameter tuning), but represent advances in artificial intelligence and deep learning that may offer greater accuracy. Therefore, we consider them to be important competitors as well. Specifically, we selected the following methods for our comparative performance analysis:

- **Temporal pixel replacement [3].** Here we perform mosaicking by copying f_{ij} for all $o_{ij} \in C$:

$$\hat{t}_{ij} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ f_{ij}, & \text{otherwise} \end{cases} \tag{8}$$

  Temporal pixel replacement is similar to the frequently used mosaicking setting 'LeastCloudy' in popular Earth observation data frameworks, such as Google Earth Engine [54]. Temporal replacement is

a method highly reliant on the availability of recent cloud-free feature images, although in practice, such recent feature data will often not be available. Although we did explore an approach incorporating basic histogram matching [4], we found the original version of the method to perform better numerically on the atmospherically corrected level 2A images used in our experiments. Moreover, other types of mosaicking, such as taking the median of the most recent cloud-free values, would require more data than other methods have access to (a time-series of past data). As a result, we selected the original temporal replacement algorithm from Equation 8 as the representative method for this approach.

- **AutoML regression ensembling**. Many machine learning algorithms can be used for regression tasks, and can be combined using ensembling to further boost performance. To ensure that the best ensemble model is configured in our experiments, we remain agnostic about the type of models in question (for example, linear regression, support vector machines or gradient boosting) and their hyperparameters, and instead automate this process using the AutoML system auto-sklearn [55]. In AutoML, the choice of machine learning model and the optimisation of its hyperparameter settings are automated, resulting in more specific, fine-tuned models over general models. In the case of auto-sklearn, multiple machine learning models are optimised using Bayesian optimisation [56], and subsequently combined into an ensemble. The available regression models include Gaussian processes, adaboost and random forests, as well as neural networks in the form of multi-layer perceptrons (MLPs). If we denote the ensemble found by auto-sklearn as $\mathcal{E}$, we compute $\hat{t}_{ij}$ as:

$$\hat{t}_{ij} = \begin{cases} o_{ij}, & \text{if } o_{ij} \notin \mathcal{C} \\ \mathcal{E}(f_{ij}), & \text{otherwise} \end{cases} \tag{9}$$

  The Auto-sklearn model was trained on the available data per patch to counteract generalisation problems, as well as to ensure that this method has access to the same amount of data as the other methods.

- **Modified Neighborhood Similar Pixel Interpolator (MNSPI) [34, 35]**. NSPI is an interpolation method originally created for the gap-filling of the relatively small gaps of Landsat 7 ETM+ data [34]. In this method, a variable spatial window is used around a missing pixel, computing the target pixel value as a weighted sum of the values of similar pixels. NSPI combines a spectro-spatial prediction, based on the spectral similarity between pixels in the same image, with a spectro-temporal prediction, based on the spectral difference in a cloud-free reference image. The method was later modified to be applied to thick cloud removal for Landsat imagery. As an interpolation method, MNSPI may perform worse on larger gaps, particularly if the gaps are larger than the maximal spatial window, whereas making the window overly large would render the algorithm computationally infeasible. In our experiments, we used a maximum window size of 17 pixels, as suggested in the original papers [34, 35].

- **DSEN2-CR [17] and UnCRtain-TS [42]** (deep learning). We opted to also explore the effectiveness of deep learning techniques specifically designed for cloud removal tasks, since this type of method is most commonly explored in recent publications, boasting impressive performance. To our knowledge, no cloud removal neural networks currently exist that are specifically aimed at cloud removal using a past cloud-free reference image, and our own preliminary explorations into effectively adapting a network to such data proved to be challenging and out of the scope of this work. Nonetheless, since these methods represent to a large degree the state of the art in cloud removal in recent years, we decided to compare VPint2 to the performance of the popular deep learning-based DSen2-CR [17] model, which leverages SAR-optical data fusion, and UnCRtain-TS, which is a multi-temporal model (also using SAR data), but can be used for a single time step. For both methods, we used the official code repositories made available by the original authors, with adaptations to the data loading procedure to load the SEN2-MSI-T dataset. This comparison did entail extra acquisitions of SAR data and ran on level 1C input data instead of level 2A, meaning the comparison between these models and VPint2 could only be performed on a separate experiment with level 1C targets, as explained in Section 5.4. Since both DSen2-CR and UnCRtain-TS were originally proposed for data fusion- and multi-temporal cloud removal, respectively, this comparison can shed light on whether such models could be successfully applied to this problem setting as well.

5.4. Experimental setup

The general approach of our experiments was as follows. First, for every patch in SEN2-MSI-T, we transferred the cloud mask to the target image as missing values, allowing us to simultaneously have access to realistic cloudy images and ground truth values, providing a middle ground between synthetic and real-world dataset evaluation. Second, we ran all methods on all scenes and their patches with all available feature sets (1 week, 1 month, 3 months and 6 months), and saved the reconstructed images as three-dimensional arrays. Following the standard of existing work [17, 42], input values were clipped to 10 000. For our analysis, we also dropped combinations of images where the alignment was incorrect, and patches on the edges of the swath where part of the patch contained no data. To compare against DSen2-CR and UnCRtain-TS,

we ran additional experiments with VPint2 and these neural networks on the level 1C (L1C) version of the target image, along with recent SAR acquisitions (though this was only used by the neural networks). In these experiments, we simulated realistic clouds using Satellite Cloud Generator [57], which the original authors found to be suitable for DSen2-CR, using cloud cover percentages sampled from the real cloud cover percentages in the main dataset. Five scenes did not have recent SAR data available and were therefore not used in this experiment. Similarly, we dropped scenes where the alignment between the L1C targets and level 2A (L2A) feature images was imperfect, and patches that the SAR data product did not cover. As a result, the dataset and sample size we used for this experiment was substantially smaller than what we used for our main experiments. The L1C and SAR data products are included in our dataset specification, and were resampled to a $10m^2$ resolution and collocated using the SNAP tool by the European Space Agency. All data used by DSen2-CR and UnCRtain-TS were preprocessed as described in the respective publications [17, 42]. The feature dataset used by VPint2 consisted of the reference image from 1 month before the target, still at a L2A processing level, but resampled to $10m^2$ to match the L1C targets. We should note, though, that this is not the ideal use case for VPint2, as it was designed to be used with the same type of data. Cloud probability masks were obtained from the Satellite Cloud Generator model directly, removing cloud detection quality as a variable, and the derived binary masks were buffered in 5 passes (5 iterations of considering pixels next to currently cloudy pixels in the mask as cloudy).

Additionally, we performed an experiment on SEN12MS-CR-TS, as described in Section 5.2. For every patch in the dataset, we loaded all 30 time steps and computed their cloud masks using the SEnSeIv2 cloud detection algorithm [26]. We then checked whether there was any combination of time steps where a cloud-free target image followed a cloudy input image, with a cloud-free feature image available at some point in the past. For all patches where these conditions were met, we ran VPint2 on the input image to create a cloud-free reconstruction, and evaluated using the cloud-free next time step.

For our numerical evaluation, we utilised several performance metrics. Firstly, we used mean absolute error (MAE):

$$MAE(\hat{\mathbf{T}}, \mathbf{T}) = \frac{1}{|\mathbf{T}|} \cdot \sum_{\hat{t}_{ij} \in \hat{\mathbf{T}}} |\hat{t}_{ij} - t_{ij}| \tag{10}$$

Secondly, we were interested in the utility of the images produced by different cloud removal methods for downstream tasks. We therefore computed the MAE on a normalised difference vegetation index (NDVI) computation task:

$$NDVI(\mathbf{x}_{ij}) = \frac{\mathbf{x}^8_{ij} - \mathbf{x}^4_{ij}}{\mathbf{x}^8_{ij} + \mathbf{x}^4_{ij}} \tag{11}$$

Here, $\mathbf{x}_{ij}$ represents a one-dimensional vector containing the band dimension of a single pixel at index ij, and the band superscripts correspond to the near-infrared (band 8) and red (band 4) bands in Sentinel-2 images (different sensors may require different bands). We then computed the MAE on the NDVI as the MAE between an NDVI derived from the reconstructed image $\hat{\mathbf{T}}$ and an NDVI derived from the ground truth image $\mathbf{T}$:

$$MAE^V(\hat{\mathbf{T}}, \mathbf{T}) = \frac{1}{|\mathbf{T}|} \sum_{\hat{t}_{ij} \in \hat{\mathbf{T}}} |NDVI(\hat{t}_{ij}) - NDVI(t_{ij})| \tag{12}$$

Thirdly, we included mean absolute percentage error (MAPE), as this gives an indication of errors regardless of the range of the underlying data, which varied between patches and between the bands within a patch:

$$MAPE(\hat{\mathbf{T}}, \mathbf{T}) = \frac{1}{|\mathbf{T}|} \cdot \sum_{\hat{t}_{ij} \in \hat{\mathbf{T}}} \frac{|\hat{t}_{ij} - t_{ij}|}{t_{ij}} \tag{13}$$

Finally, we included the structural similarity index measure (SSIM) [58] as an indication of the quality of the produced images in terms of human perception:

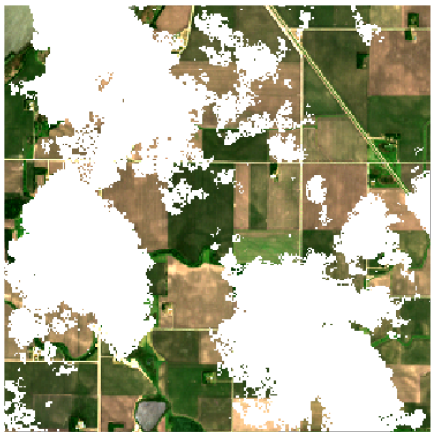
$$SSIM(\hat{\mathbf{T}}, \mathbf{T}) = \frac{(2 \cdot \mu_{\hat{\mathbf{T}}} \cdot \mu_{\mathbf{T}}) \cdot (2 \cdot \sigma_{\hat{\mathbf{T}}\mathbf{T}} + c_2)}{(\mu^2_{\hat{\mathbf{T}}} + \mu^2_{\mathbf{T}} + c_1) \cdot (\sigma^2_{\hat{\mathbf{T}}} + \sigma^2_{\mathbf{T}} + c_2)} \tag{14}$$

Here, μ and σ represent the mean and standard deviation of the pixel values of the given image, respectively, and c_1 and c_2 are constants, for which we used the default values $c_1 = (0.01 \cdot L)^2$, $c_2 = (0.03 \cdot L)^2$ (where $L = \max(\mathbf{T}) - \min(\mathbf{T})$).

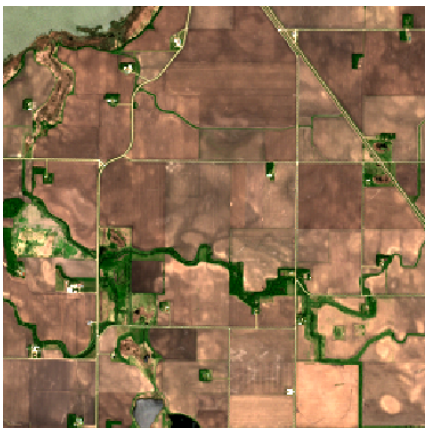
Additionally, our experiments on SEN12MS-CR-TS included the peak signal-to-noise ratio (PSNR), root mean squared error (RMSE) and spectral angle mapper (SAM) performance metrics, as these are the metrics methods are compared to in the original UnCRtain-TS paper [42]. We computed the metrics using the implementation provided with the code of UnCRtain-TS.

6. Results and discussion

A visual example of a SEN2-MSI-T patch, with its feature image, cloudy version and example reconstructions by the different methods, can be found in Figure 4. The bottom row contains results for VPint2 for a full scene reconstruction that was not split into patches, showing that VPint2 can be applied to larger images as well. In the figure, the reconstruction by temporal replacement contains outdated information for the vegetation in the target image. AutoML regression, due to conflicting relationships between pixels in the old image and the new, where some contain more vegetation in the target image while other pixels remained similar, ended up predicting mainly a mean value somewhere in between. MNSPI created blurry and occasionally outdated (similar to temporal replacement) visually plausible results within its spatial

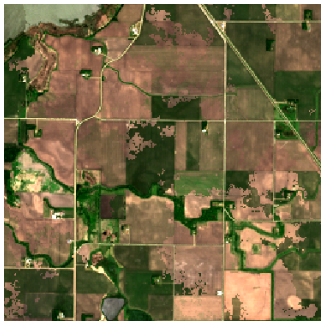


(a) Input

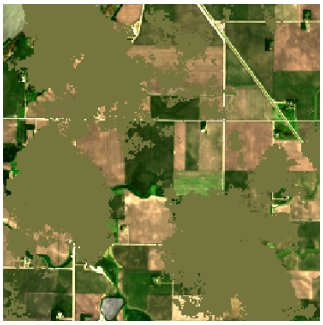


(b) Feature

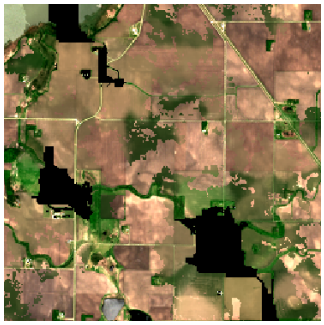
(c) Target



(d) Replacement

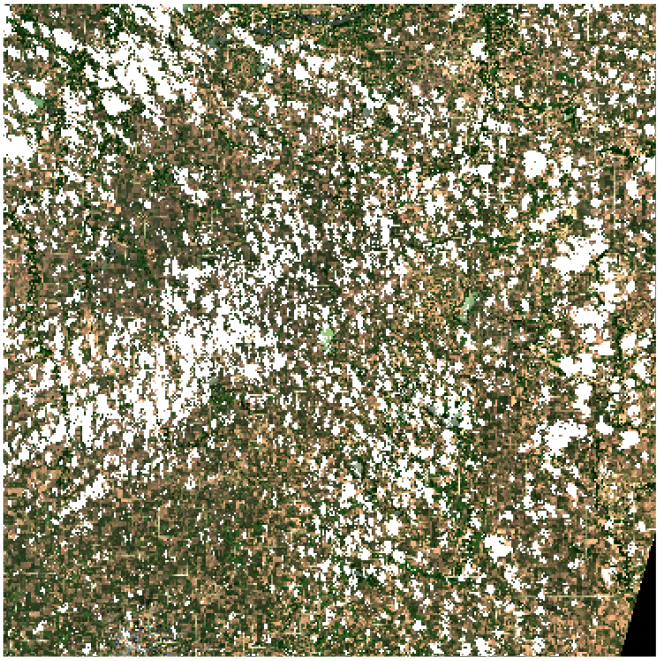


(e) AutoML regression



(f) MNSPI

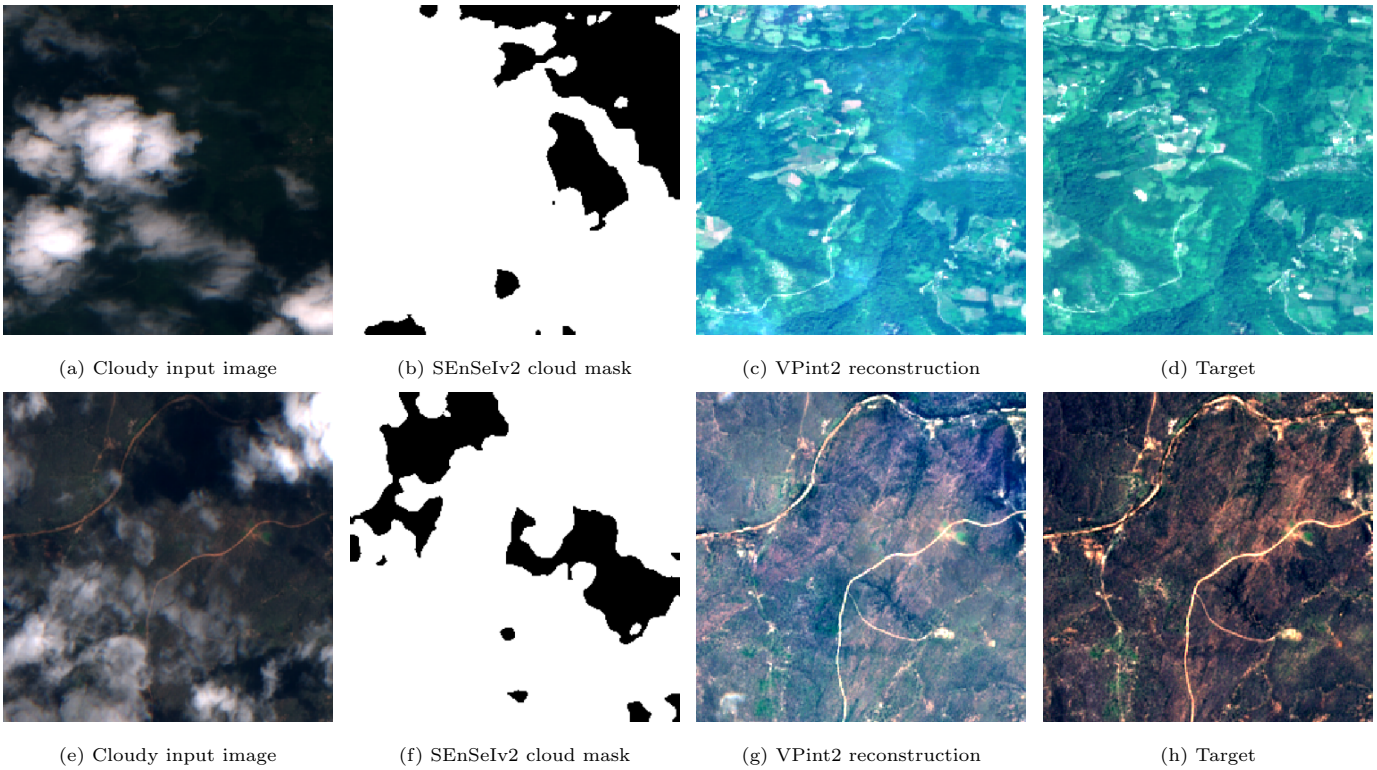
(g) VPint2



(h) Input (full scene)

(i) VPint2 (full scene)

Figure 4: Example reconstruction visualisations for VPint2, temporal replacement, AutoML regression and MNSPI. The visualised patch originated from a scene in Iowa, USA, with a cropland land cover class. The top row shows the feature-, target- and input images, the middle row shows the reconstructions by the different methods, and the bottom row shows the input and reconstruction of a full-sized scene.



(a) Cloudy input image (b) SEnSeIv2 cloud mask (c) VPint2 reconstruction (d) Target

(e) Cloudy input image (f) SEnSeIv2 cloud mask (g) VPint2 reconstruction (h) Target

Figure 5: Two example reconstructions with an average performance by VPint2 on the SEN12MS-CR-TS dataset (5c, 5g), along with the input images (5a, 5e), cloud mask (5b, 5f; white pixels denote clouds or cloud shadow) and the temporally close target image (5d, 5h). The cloud-free regions of the input images are dark in the visualisation due to the relatively high reflectance values of the clouds.

window, but failed to make a prediction for pixels outside of its window (while increasing this window further would render it computationally infeasible). The reconstruction by VPint2, seems the most visually plausible and seems to contain the most up-to-date information out of these methods on this example patch, with its greatest visual weakness appearing to be the propagation of incorrect colours in small parts of the image. Two visual examples of SEN12MS-CR-TS patches, with cloudy inputs and their cloud masks, VPint2 reconstructions, and the cloud-free target images of the next time step, can be found in Figure 5. In this figure, the reconstructions appeared visually plausible, although the hue of the images were different between the VPint2 reconstruction and the target image. This was likely caused by a difference in atmospheric conditions between the Level 1C feature- and target images, and supports our intuition that a use case on atmospherically corrected Level 2A images would be preferred.

In the following, we report the results for specific questions in detail.

6.1. Q1: Identity priority and elastic band resistance

To answer Q1, we investigated the effect of the extensions we made to the VPint2 algorithm as described in Section 4.3. These extensions were identity priority, aimed at reducing the impact of artefacts appearing on the edges of objects due to temporal heterogeneity, and elastic band resistance, aimed at preventing an explosion of extremely high values caused by, for example, quality issues in the data. To gauge the impact of these extensions, we performed an ablation experiment by running the main experiments for VPint2 three times: once with all features enabled, once with identity priority disabled (denoted as 'no ID'), and once with elastic band resistance disabled (denoted as 'no EB'). These results can be found in Table 1 and show that the added value of our extensions depends on the land cover type.

Identity priority appears to be particularly effective at improving performance on the urban- and cropland scenes. This is in line with expectations, since urban areas contain many smaller objects for which the between-object spatial relationship may change (such as bright reflections on roofs), resulting in temporal heterogeneity that can be alleviated using identity priority. Similarly, on cropland, the growth and harvest cycles may not have been applied uniformly to all fields, resulting in temporally heterogenous between-object spatial relationships. Elastic band resistance appears to be important on the cropland scenes, but not significantly different from the normal VPint2 results on other land covers. This further underlines that exploding values are rare, but if the phenomenon does occur, performance can be significantly improved by enabling this enhancement. Therefore, elastic band resistance can be an important tool for cloud removal on certain scenes,

Table 1: Numerical results of our experiments. The best performing method per metric, where ↓ indicates a measure to be minimised and ↑ indicates a measure to be maximised, was computed using a one-sided Wilcoxon signed-rank test at a significance level $\alpha = 0.05$, and has been marked **bold** (ties allowed).

	VPint2	**VPint2 (no IP)**	**VPint2 (no EB)**	**Replacement**	**AutoML regression**	**MNSPI**
			Cropland			
↓ MAE	**357.00±291.72**	367.65±303.88	370.11±310.79	614.23±450.28	460.19±293.43	**363.15±331.40**
↓ MAE^V	**0.0918±0.0787**	**0.0929±0.0799**	**0.0921±0.0809**	0.1563±0.1145	0.1239±0.0765	**0.0995±0.0983**
↓ MAPE	**24.398±58.405**	25.028±58.235	25.511±59.640	27.258±29.552	28.979±61.704	**27.277±63.188**
↑ SSIM	**0.8378±0.1907**	0.8317±0.1936	0.8332±0.1954	0.7028±0.2562	0.7994±0.1889	0.8021±0.2368
			Forest			
↓ MAE	**199.29±227.73**	**206.64±253.55**	**199.19±223.86**	421.39±473.35	402.28±620.58	**214.35±341.47**
↓ MAE^V	**0.0374±0.0274**	**0.0377±0.0282**	**0.0368±0.0256**	0.0758±0.0691	0.0857±0.0971	0.0494±0.0520
↓ MAPE	**6.442±4.508**	**6.620±5.422**	**6.469±4.676**	14.683±14.343	12.183±9.188	**7.003±8.109**
↑ SSIM	**0.9445±0.0705**	**0.9404±0.0855**	**0.9448±0.0712**	0.8756±0.1586	0.8731±0.1599	0.8662±0.2172
			Herbaceous			
↓ MAE	192.91±125.64	**191.18±128.91**	193.45±127.75	549.80±1392.8	264.27±145.17	222.86±201.32
↓ MAE^V	0.0361±0.0451	**0.0359±0.0463**	**0.0354±0.0450**	0.0476±0.0552	0.0394±0.0273	0.0426±0.0477
↓ MAPE	7.624±6.015	**7.543±6.080**	7.621±6.093	22.021±54.523	10.280±6.510	9.271±8.738
↑ SSIM	**0.9400±0.0881**	**0.9397±0.0947**	**0.9404±0.0896**	0.8718±0.2988	0.9321±0.0705	0.8397±0.1873
			Shrubs			
↓ MAE	**162.40±153.25**	**160.95±150.84**	**163.03±155.49**	315.76±202.21	286.42±235.04	209.88±229.97
↓ MAE^V	**0.0229±0.0273**	**0.0233±0.0290**	**0.0226±0.0271**	0.0379±0.0452	0.0293±0.0308	0.0312±0.0477
↓ MAPE	**8.780±81.345**	**9.680±108.487**	**9.595±100.018**	12.647±22.941	10.977±9.593	12.064±117.163
↑ SSIM	**0.9661±0.0599**	**0.9646±0.0632**	**0.9663±0.0602**	0.9466±0.0658	0.9427±0.0819	0.8357±0.1873
			Urban			
↓ MAE	**314.58±183.26**	335.39±201.86	**318.48±199.24**	553.24±494.50	590.15±390.03	388.52±231.95
↓ MAE^V	**0.1057±0.0782**	0.1166±0.0903	**0.1054±0.0784**	0.1465±0.1415	0.1487±0.0546	0.1244±0.1301
↓ MAPE	**18.421±18.116**	21.635±30.842	**19.106±21.157**	46.597±99.988	32.683±36.395	22.264±28.840
↑ SSIM	**0.7857±0.1575**	0.7695±0.1692	0.7755±0.1772	0.7180±0.2386	0.6674±0.1660	0.6563±0.2764

but will not be necessary for most other problem settings.

In conclusion, the enhancements of VPint2, appear to improve the performance of VPint2 when enabled, although they are mainly necessary on specific land cover classes. Since VPint2 performed significantly better than the versions without our extensions in these cases, while the results for VPint2 on other land cover classes was generally not significantly worse than those without the extra functionalities, we will report the results for VPint2 with both enhancements enabled in subsequent experiments.

6.2. Q2: Comparative analysis

The numerical results of our empirical performance comparison on SEN2-MSI-T can be found in Table 1, and the distribution of the performance of the different methods has been visualised per land cover class in Figure 6. As the table and figure show, VPint2 achieved an improvement in performance over temporal replacement, AutoML regression and NSPI in all cases, which was statistically significant in all cases but the comparison with NSPI on cropland and forest land cover classes. The relatively similar performance of NSPI on these two land cover classes may indicate that the use of local spatial information in the input image, which both VPint2 and MNSPI exploit, is especially important for land cover with more vegetation. The spread of the performance by VPint2, as seen in Figure 6, tended to be smaller as well.

All methods performed worse on the cropland and urban scenes compared to other land covers, reflecting their challenge as dynamic land cover types (both in terms of values and spatial structure). AutoML regression often ended up predicting close to the target mean value, as can be seen in Figure 4e. This was likely caused by conflicting relationships between feature and target pixels in their respective images. For example, in Figures 4b and 4c, almost all feature pixels are a similar brown, whereas the target pixels for some fields were deep green, and some were light brown. As a result, in these cases, the algorithm appears to have converged to models predicting the mean, since the pixel-wise feature-to-target relationships were inconsistent and conflicting.

We conclude that VPint2 would be a better cloud removal method than temporal replacement, AutoML regression and MNSPI in the majority of cases.

The results for the experiment comparing against DSen2-CR and UnCRtain-TS on L1C targets can be found in Table 2. In this experiment, VPint2 performed better than UnCRtain-TS, and better than DSen2-CR on almost all land cover classes, with the exception of MAE^V on cropland and urban land covers, and a non-significant

Table 2: Numerical results of our experiments on L1C data for the subset of SEN2-MSI-T for which additional Sentinel-1 SAR data was available. The best performing method per metric, based on a one-sided Wilcoxon signed-rank test at significance level $\alpha = 0.05$, has been marked in **boldface**.

	VPint2	**DSen2-CR**	**UnCRtain-TS (single time step)**
		Cropland	
↓ MAE	**189.88±60.90**	218.95±60.58	734.71±364.02
↓ MAE^V	0.0650±0.0271	**0.0445±0.0275**	0.1260±0.1429
↓ MAPE	**20.254±9.273**	51.780±49.219	28.338±17.161
↑ SSIM	**0.9182±0.0704**	**0.9133±0.0548**	0.3180±0.4199
		Forest	
↓ MAE	**163.67±43.35**	939.84±68.69	1109.16±342.80
↓ MAE^V	**0.0357±0.0126**	0.1424±0.0225	0.2875±0.0634
↓ MAPE	**6.257±1.923**	41.174±2.808	53.265±7.515
↑ SSIM	**0.9760±0.0169**	0.8145±0.0506	0.5667±0.3323
		Herbaceous	
↓ MAE	**182.07±64.27**	757.43±116.02	1147.27±342.80
↓ MAE^V	**0.0142±0.0070**	0.0634±0.0221	0.1241±0.1106
↓ MAPE	**7.106±3.107**	31.502±3.303	47.309±12.217
↑ SSIM	**0.9650±0.0308**	0.8484±0.0500	0.7254±0.1210
		Shrubs	
↓ MAE	**159.29±60.51**	684.01±343.74	1016.05±521.61
↓ MAE^V	**0.0225±0.0194**	0.0332±0.0361	0.1171±0.1086
↓ MAPE	**7.182±3.592**	30.066±11.997	39.309±14.875
↑ SSIM	**0.9704±0.0263**	0.8877±0.0573	0.8026±0.1238
		Urban	
↓ MAE	**182.22±53.50**	327.81±95.03	512.64±318.79
↓ MAE^V	0.0379±0.0211	**0.0282±0.0222**	0.0784±0.0832
↓ MAPE	**12.756±5.370**	36.782±32.532	23.576±13.883
↑ SSIM	**0.9628±0.0234**	0.9272±0.0414	0.8817±0.1030

Table 3: Comparison of the results of VPint2 on **a subset of** SEN12MS-CR-TS, against the multi-temporal performance of methods on the **full dataset** reported by Ebel et al. [42] Since this comparison is only for reference, and the methods were evaluated on different parts of the dataset, we do not mark the best performance.

Method	**↓ RMSE**	**↑ PSNR**	**↑ SSIM**	**↓ SAM**
DSen2-CR	0.079	26.04	0.810	12.147
STGAN	0.060	25.42	0.818	12.548
CR-TS Net	0.057	26.68	0.836	10.657
U-TAE	0.051	27.05	0.849	11.649
UnCRtainTS	0.051	27.84	0.866	10.160
VPint2 (suitable subset of data)	0.042	30.38	0.928	6.541

improvement on SSIM for cropland. The performance of VPint2 was stronger in this experiment for SSIM on urban and cropland scenes, in particular, compared to the main experiments from Table 1. This was likely caused by the use of 1 month-old feature images boosting performance, although the relatively low contrast on L1C targets compared to L2A products may have also played a role.

The comparison to DSen2-CR is generally favourable for VPint2, despite being used outside of its intended application of using the exact same type of feature data, and the mean absolute error for DSen2-CR was higher than expected given the performance reported in Meraner et al. [17] for the forest, herbaceous and shrubs scenes. However, the (value-independent) SSIM was on par, or sometimes better than, what was reported in this paper. As a result, the higher MAE may have been caused by the range of the values themselves in different land cover

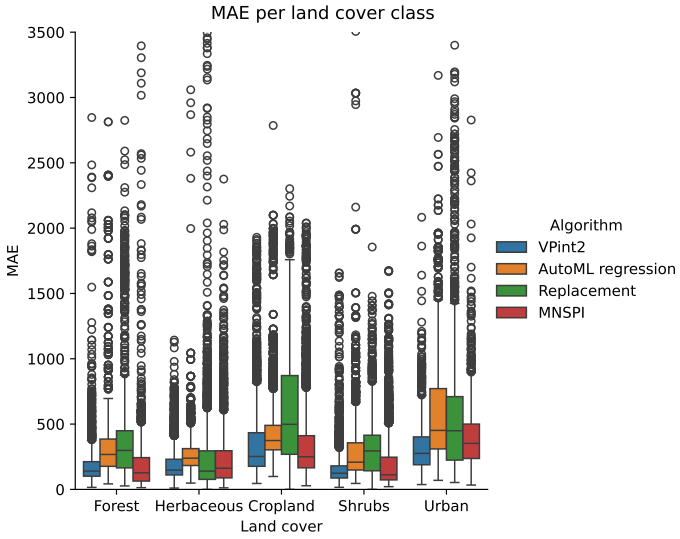


Figure 6: Box plots of the distributions of error rates (MAE) of the different methods for the five land cover classes. The visualisation has been limited to error rates of 3000, due to the outliers of the different methods reducing the legibility of the plots.

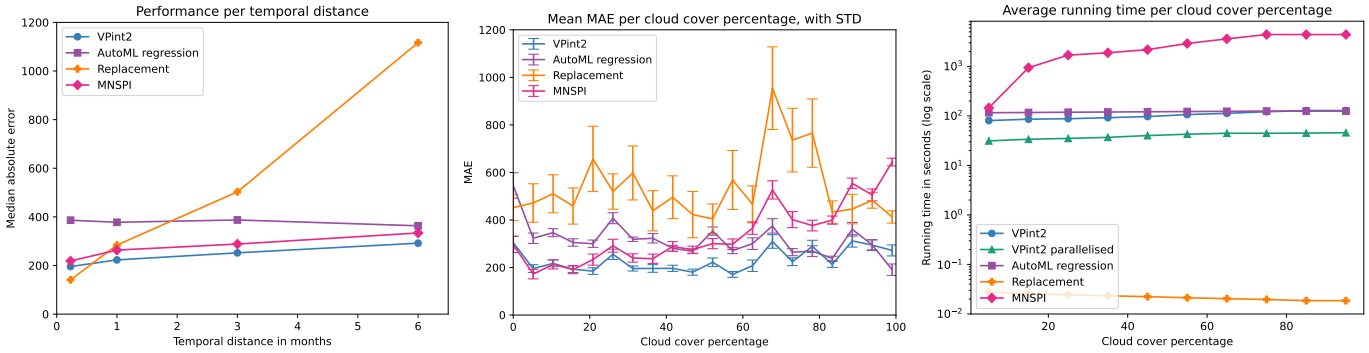
classes, rather than a truly worse performance, especially considering the relatively similar MAPE for all the land cover classes. UnCRtain-TS, a method intended for multi-temporal cloud removal, had the highest error rates and greatest variation in performance. A possible cause for this behaviour may lie in differences between the dataset it was trained on and our benchmark dataset.

Addressing this type of concern, the results for our experiment on SEN12MS-CR-TS, shown in Table 3, demonstrate that VPint2 performs very well on this dataset, in comparison with existing methods on a task they were designed for. We stress that this comparison is mainly included to put the results of our proposed method in perspective compared to a majority of recent state-of-the-art methods; since our method is only suitable for a subset of problem instances in the dataset, these results cannot be used to conclude that one method performs better than another, as competing methods might have also performed better on this subset. However, the results do indicate that, on a subset of suitable instances, there can be numerical advantages to using VPint2 over existing methods.

6.3. Q3 and Q4: Patch properties and computational efficiency

To answer Q3, we plotted the relationship between the temporal distance of the feature image and the percentage of cloud cover in the target image, with the performance achieved by the different methods, while for Q4, we also plotted the average (out of 20 randomly selected patches per level) end-to-end running time in seconds for the different methods for different levels of cloud cover. We show these visualisations in Figure 7.

In the case of temporal distance, the results are as expected: Temporal replacement performs very well for tem-



(a) Temporal distance (b) Cloud cover percentage (c) Running time

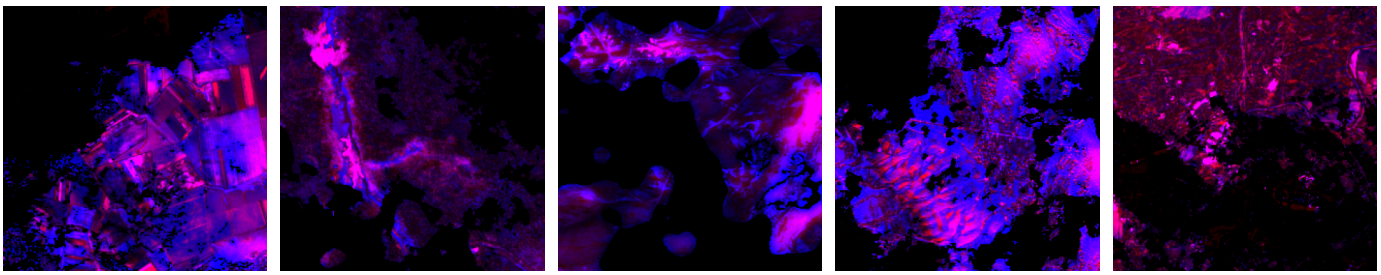
Figure 7: The sensitivity of different methods to the temporal distance of the feature image (7a), the cloud cover percentage of the input image (7b), based on mean absolute error (MAE), and the average running time in seconds out of 20 random patches per cloud cover percentage (7c). For Figure 7b, the figure was created by computing the average error for whole percentages and smoothing the resulting curve using splines interpolation. We added error bars for the standard deviation, to maintain an indication of the variability of results. The peak in errors for temporal replacement, which is a pixel-based method and should not be affected by the cloud cover percentage, was likely caused by the exceptionally large errors on particular problem instances (such as snowed-over herbaceous scenes at a distance of 6 months), that happened to contain a relatively large cloud cover percentage.

poral distances of 1 week, but quickly loses its effectiveness as the temporal distance increases. VPint2 and MNSPI are also affected by the temporal distance, presumably due to cases where the spatial structure of a scene was altered over time, but the effect is fairly mild. AutoML regression appears to not be affected by this variable, though slightly lower errors can be observed for larger temporal distances. However, this effect is small enough (386.27 at 1 week, 363.36 at 6 months), that this was likely caused by chance, rather than a true pattern. The temporal distance was especially important for herbaceous land cover scenes, which were exceptionally static on shorter temporal distances, but also changed exceptionally strongly for longer temporal distances due to seasonal effects (mainly snow cover).

The results in Figure 7a carry implications for the evaluation approach of cloud removal methods. In our experiments, we transferred cloud masks from a cloudy image at the same location as the target image, giving us access to realistic cloud cover as well as real ground truth values. On the other hand, many of the popular real-world cloud removal datasets used in high-profile work, such as SEN12MS-CR [17] and SEN12MS-CR-TS [51], rely on evaluating (and training) models by treating a co-located recent cloud-free acquisition as ground truth (with the closest possible time interval for Sentinel-2 being one 5-day revisit). Similarly, the feature image with a temporal distance of 1 week in our dataset consisted of 1 (preferred) or 2 (if necessary) revisits, which temporal replacement mosaicked into the target image as a cloud removal method. Therefore, the results for temporal replacement at 1 week in Figure 7a are an indication of the reliability that could be expected of real-world datasets. Although temporal replacement performed better at this temporal distance than other methods, its MAE at 1 week (140.97) reached levels comparable to the magnitude of the errors of VPint2 for all tested temporal distances (195.64 to 291.82).

These results suggest that, when using a purely real-world evaluation approach, the magnitude of the aleotoric uncertainty of the dataset would be comparable to the magnitude of the performance of cloud removal methods themselves, resulting in noisy and potentially unreliable evaluation. We therefore recommend further research to consider using a cloud mask transfer-based approach, as we have employed in SEN2-MSI-T, to evaluate cloud removal methods more reliably. Although this may not have been possible for neural networks, which do not use explicit cloud masks, and must therefore represent clouds realistically in the input image, recent advances in cloud simulation [57] may allow even neural networks to be trained on data with true ground-truth values available.

In the case of cloud cover percentage, a few observations can be made. First, VPint2 was not as heavily affected by larger percentages as might be expected from interpolation methods based on previous work [15]. Although there is an increase in errors (and variability) from 60% to 100%, there is no point where VPint2 clearly performs worse than the alternative methods apart from 100% cloud coverage. Second, temporal replacement was not affected by this variable, with the exception of a spike between 60% and 80%. Since this method is purely pixel-based, it is unlikely that a property of these particular clouds caused the spike. Instead, it is more likely that the performance on a specific challenging condition (for example, herbaceous land cover at 6 months) simply contained more patches with large clouds. Third, VPint2 and AutoML regression contain similar spikes in error rates, implying that similar patches are more challenging for both methods. However, VPint2 had lower error rates than AutoML regression, except for the highest cloud cover percentages, where performance was highly similar. Finally, MNSPI achieved



(a) Cropland (b) Forest (c) Herbaceous (d) Shrubs (e) Urban

Figure 8: Visualisation of reconstruction errors (normalised and scaled to a 0-1 range) by VPint2 (red) and AutoML regression (blue) for an example patch from every land cover class, with a purple colour indicating overlapping errors. The existence of areas with mostly blue or red colours, as opposed to a constant purple colour, indicates complementary strengths between the two methods.

results comparable to VPint2 for low cloud cover percentages, but its MAE increased steeply for higher cloud cover percentages. Overall, it appears that VPint2 is effective at addressing this weakness of interpolation methods, with the caveat that its competitive advantage over competing (non-interpolation) methods does slightly decrease for very high percentages, where it performs on par with the most competitive alternative method.

In terms of running time, Figure 7c shows that temporal replacement was by far the fastest method, with a running time on the order of magnitude of 0.01 throughout. VPint2 was the second fastest method, with its parallelised version reducing the average running times from about 100 seconds to between about 30 to 40 seconds. A mild increase in running time can be observed as the cloud cover percentage increases. MNSPI had a running time comparable to the serial version of VPint2 for low cloud cover, but rose to a running time exceeding 1000 seconds after about 20% cloud cover, likely caused by the need for larger window sizes. AutoML regression had a running time slightly above 1000 seconds; however, this includes training as well as algorithm selection and hyperparameter optimisation time, which was limited by a user-supplied parameter. Therefore, since its inference impact is negligible, its running time depends mainly on how long a user will allow it to search for good configurations (but a lower budget may result in worse numerical performance).

6.4. Q5: Complementary strengths and ensembling

Although VPint2 achieved strong performances in our experiments, as seen in Table 1, a method that performs best on average is not necessarily the strongest on all instances. This is especially the case for Earth observation data, which is inherently diverse in terms of sensors, spectral bands, landscape, atmospheric conditions and more. As a result, when evaluating new algorithms applicable to this type of data, it is potentially problematic to merely consider average performance over a diverse collection of datasets. Instead, we believe that it is preferable to assess the relative strength of new approaches on individual datasets, and in particular, to focus algorithm development on scenarios where currently available methods appear to perform relatively poorly.

From this perspective, when comparing the performance of different methods, it is important to assess the complementarity of the strengths of the methods. We therefore visualised the reconstruction errors for VPint2 *vs* AutoML regression in Figure 8. In this figure, VPint2 errors were visualised in the red band, and AutoML regression errors were visualised in the blue band, meaning that only regions with a purple colour would show a strong overlap in performance. Since Figure 8 shows many regions with either red or blue colours, it is clear that both methods have strengths that the other does not.

We explored this idea further by probing the potential of ensembling approaches. Although a fully functional ensembling approach would entail addressing non-trivial challenges, such as finding informative features for automated algorithm selection, and is therefore beyond the scope of this work, we wish to show the potential of this type of approach using an "oracle"-based experiment. We show the results in terms of MAE that can be achieved in this way in Table 4; we only considered MAE, since the ranking of the methods we studied was consistent across all performance metrics.

The experiments whose results are reported in the table were carried out as follows. First, we ran preliminary experiments using an "oracle" ensemble on a pixel level, selecting the most accurate predicted value out of VPint2, AutoML regression and temporal replacement, for every pixel. This produces a lower bound of the error rates achievable by a perfect ensemble. As shown in Table 4, this approach consistently significantly outperforms the best individual method for every scene, which demonstrates that in principle, an ensemble (when accurately selecting methods) could achieve substantially better results than any single method. We also ran this oracle setup without including VPint2 and observed a significantly reduced performance, further underlining the degree to which VPint2 contributes to the state of the art in cloud removal. Finally, we included a setup selecting methods per patch instead of per pixel, which could be used to assess whether the complementary strengths of methods occurred at the

Table 4: Numerical results of our ensembling explorations (MAE only). The oracle approaches could not be used in practice, and serve as a lower bound of what a perfect ensemble could achieve with these methods. The best individual method is shown for every land cover class as reference, and corresponds to the strongest method for that land cover class in Table 1.

	Cropland	Forest	Herbaceous	Shrubs	Urban
Oracle pixel	**295.19±342.45**	**152.36±246.99**	**180.44±161.64**	**178.14±240.31**	**277.34±210.43**
Oracle pixel (no VPint)	308.36±326.58	169.70±253.35	182.67±151.80	189.76±230.23	312.70±211.84
Oracle patch	344.10±310.40	208.26±268.57	164.42±132.12	182.22±192.00	317.78±208.48
Oracle patch (no VPint)	406.90±298.48	259.37±270.76	183.82±136.31	244.56±183.21	344.47±219.70
Best individual method	357.00±291.27	199.19±223.86	191.18±128.91	160.95±150.84	314.58±183.26

patch- or pixel-level. The results for patch-level ensembling were much closer to the best individual method. This suggests that a perfect ensemble nearly always selects the same method for every patch as the overall best method for the land cover class, and that properties of the patch do not contribute strongly toward which method performs best. Moreover, we observe that complementary strengths occur at pixel- rather than at patch-level.

7. Concluding remarks and future directions

In this work, we have extended the spatial interpolation algorithm VPint [5] to create VPint2, which is aimed at addressing optical remote sensing cloud removal problems. We made four key technical contributions to the original VPint algorithm, namely the use of exact weights computed directly from co-located past imagery, a running time speedup using parallel computing over bands, identity priority and elastic band resistance, addressing the temporal heterogeneity and exploding values problems in remote sensing data and allowing VPint2 to perform well on optical Earth observation data. Our proposed method does not use any additional data compared to temporal replacement, and requires no training procedure. It also automatically adapts its parameters to the best values based on the available data, which is necessary, because the appropriate settings can vary greatly even within a single image. We created an evaluation benchmark dataset called SEN2-MSI-T, consisting of 20 geographically diverse scenes for the five most common land cover classes, enabling us to evaluate cloud removal methods on a diverse set of environmental conditions and spatial patterns that also provides users with true ground truth values. The results from our experiments indicate that this method of evaluation is more reliable than common approaches using fully real-world datasets. Additionally, we performed an experiment on a subset of the popular SEN12MS-CR-TS dataset to better place our work in the context of recent work. Our empirical results show that VPint2 significantly outperforms alternative methods on all land cover classes on average. We have also found that VPint2 is only mildly affected by the temporal distance of its reference image, which algorithms should be robust to as it may take several months to acquire cloud-free imagery during a rainy season, and to the percentage of cloud cover in a patch, allowing it to be applied to a wide range of cloud cover conditions. Our experiments found that VPint2 is more computationally efficient than existing cloud removal interpolation methods such as MNSPI, and our newly introduced parallelisation further cut the running time for VPint2 by about 60% to 70%. Our findings also encourage the adoption of an approach where new cloud removal methods are evaluated based on their specialist utility in a certain subset of use cases, as general methods tend to perform worse in inherently diverse domains, such as Earth observation data. Our "oracle"-based experimental results show that ensembling approaches using the strengths of multiple methods, especially on a pixel level, hold great potential for further performance improvements. We believe that identifying useful instance features for practically applicable ensembling approaches would be a fruitful endeavour in future work.

Other future work could explore the potential efficacy of VPint2 for time-series cloud removal, deriving weights from the cloud-free regions of the images in the time-series and combining these into one set of weights, or exploring the impact of adding feature data of different sensor modalities, such as SAR.

In conclusion, VPint2, as an easy-to-apply and effective cloud removal method, has shown its potential in terms of performance, as well as its complementarity with existing methods.

Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is part of the project "Physics-aware Spatio-temporal Machine Learning for Earth Observation Data" (with project number OCENW.KLEIN.425) of the research programme Open Competition ENW which is partly financed by the Dutch Research Council (NWO). This research is also partly financed by the European Space Agency (ESA) under the Open Space Innovation Platform (OSIP) research project "Physics-aware Automated Machine Learning (PA-AutoML) for Earth Observations". It was also partially supported by TAILOR, a

project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

The authors would like to extend special thanks to Athanasia Nikolaou for her contributions to this project in its early stages.

References

[1] E. El-Araby, T. El-Ghazawi, J. Le Moigne, R. Irish, Reconfigurable processing for satellite on-board automatic cloud cover assessment, Journal of Real-Time Image Processing 4 (3) (2009) 245–259.

[2] G. Giuffrida, L. Diana, F. de Gioia, G. Benelli, G. Meoni, M. Donati, L. Fanucci, Cloudscout: a deep neural network for on-board cloud detection on hyperspectral images, Remote Sensing 12 (14) (2020) 2205.

[3] M. Li, S. C. Liew, L. K. Kwoh, Producing cloud free and cloud-shadow free mosaic from cloudy ikonos images, in: IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No. 03CH37477), Vol. 6, Ieee, 2003, pp. 3946–3948.

[4] E. H. Helmer, B. Ruefenacht, Cloud-free satellite image mosaics with regression trees and histogram matching, Photogrammetric Engineering & Remote Sensing 71 (9) (2005) 1079–1089.

[5] L. Arp, M. Baratchi, H. H. Hoos, Vpint: value propagation-based spatial interpolation, Data Mining and Knowledge Discovery 36 (2022) 1647—-1678.

[6] H. Shen, H. Li, Y. Qian, L. Zhang, Q. Yuan, An effective thin cloud removal procedure for visible remote sensing images, ISPRS Journal of Photogrammetry and Remote Sensing 96 (2014) 224–235.

[7] J. Liu, X. Wang, M. Chen, S. Liu, X. Zhou, Z. Shao, P. Liu, Thin cloud removal from single satellite images, Optics express 22 (1) (2014) 618–632.

[8] G. Hu, X. Li, D. Liang, Thin cloud removal from remote sensing images using multidirectional dual tree complex wavelet transform and transfer least square support vector regression, Journal of Applied Remote Sensing 9 (1) (2015) 095053.

[9] M. Xu, X. Jia, M. Pickering, S. Jia, Thin cloud removal from optical remote sensing images using the noise-adjusted principal components transform, ISPRS Journal of Photogrammetry and Remote Sensing 149 (2019) 215–225.

[10] W. Li, Y. Li, D. Chen, J. C.-W. Chan, Thin cloud removal with residual symmetrical concatenation network, ISPRS Journal of Photogrammetry and Remote Sensing 153 (2019) 137–150.

[11] J. Li, Z. Wu, Z. Hu, J. Zhang, M. Li, L. Mo, M. Molinier, Thin cloud removal in optical remote sensing images based on generative adversarial networks and physical model of cloud distortion, ISPRS Journal of Photogrammetry and Remote Sensing 166 (2020) 373–389.

[12] J. Li, Z. Wu, Z. Hu, Z. Li, Y. Wang, M. Molinier, Deep learning based thin cloud removal fusing vegetation red edge and short wave infrared spectral information for sentinel-2a imagery, Remote Sensing 13 (1) (2021) 157.

[13] Y. Guo, W. He, Y. Xia, H. Zhang, Blind single-image-based thin cloud removal using a cloud perception integrated fast fourier convolutional network, ISPRS Journal of Photogrammetry and Remote Sensing 206 (2023) 63–86.

[14] C. Zhang, W. Li, D. Travis, Gaps-fill of slc-off landsat etm+ satellite image using a geostatistical approach, International Journal of Remote Sensing 28 (22) (2007) 5103–5122.

[15] H. Shen, X. Li, Q. Cheng, C. Zeng, G. Yang, H. Li, L. Zhang, Missing information reconstruction of remote sensing data: A technical review, IEEE Geoscience and Remote Sensing Magazine 3 (3) (2015) 61–85.

[16] USGS, Phase 2 gap-fill algorithm: SLC-off gap-filled products gap-fill algorithm methodology., `https://www.usgs.gov/faqs/what-landsat-7-etm-slc-data`, accessed: 19-01-2022 (2004).

[17] A. Meraner, P. Ebel, X. X. Zhu, M. Schmitt, Cloud removal in sentinel-2 imagery using a deep residual neural network and sar-optical data fusion, ISPRS Journal of Photogrammetry and Remote Sensing 166 (2020) 333–346.

[18] F. Xu, Y. Shi, P. Ebel, L. Yu, G.-S. Xia, W. Yang, X. X. Zhu, Glf-cr: Sar-enhanced cloud removal with global–local fusion, ISPRS Journal of Photogrammetry and Remote Sensing 192 (2022) 268–278.

[19] S. Han, J. Wang, S. Zhang, Former-cr: A transformer-based thick cloud removal method with optical and sar imagery, Remote Sensing 15 (5) (2023) 1196.

[20] H. Liu, B. Huang, J. Cai, Thick cloud removal under land cover changes using multisource satellite imagery and a spatiotemporal attention network, IEEE Transactions on Geoscience and Remote Sensing 61 (2023) 1–18.

[21] M. Xu, F. Deng, S. Jia, X. Jia, A. J. Plaza, Attention mechanism-based generative adversarial networks for cloud removal in landsat images, Remote sensing of environment 271 (2022) 112902.

[22] F. N. Darbaghshahi, M. R. Mohammadi, M. Soryani, Cloud removal in remote sensing images using generative adversarial networks and sar-to-optical image translation, IEEE Transactions on Geoscience and Remote Sensing 60 (2021) 1–9.

[23] R. Jing, F. Duan, F. Lu, M. Zhang, W. Zhao, Denoising diffusion probabilistic feature-based network for cloud removal in sentinel-2 imagery, Remote Sensing 15 (9) (2023) 2217.

[24] A. H. Sanchez, M. C. A. Picoli, G. Camara, P. R. Andrade, M. E. D. Chaves, S. Lechler, A. R. Soares, R. F. Marujo, R. E. O. Simões, K. R. Ferreira, et al., Comparison of cloud cover detection algorithms on sentinel–2 images of the amazon tropical forest, Remote Sensing 12 (8) (2020) 1284.

[25] SentinelHub, s2cloudless, `https://github.com/sentinel-hub/sentinel2-cloud-detector`, accessed: 16-03-2022 (2018).

[26] A. Francis, Sensor independent cloud and shadow masking with partial labels and multimodal inputs, IEEE Transactions on Geoscience and Remote Sensing (2024).

[27] F. Zou, Q. Hu, Y. Liu, H. Li, X. Zhang, Y. Liu, Spatiotemporal changes and driving analysis of ecological environmental quality along the qinghai–tibet railway using google earth engine—a case study covering xining to jianghe stations, Remote Sensing 16 (6) (2024) 951.

[28] J. Brandt, J. Ertel, J. Spore, F. Stolle, Wall-to-wall mapping of tree extent in the tropics with sentinel-1 and sentinel-2, Remote Sensing of Environment 292 (2023) 113574.

[29] P. Scaramuzza, J. Barsi, Landsat 7 scan line corrector-off gap-filled product development, in: Proceeding of Pecora, Vol. 16, 2005, pp. 23–27.

[30] H. Feilhauer, G. P. Asner, R. E. Martin, S. Schmidtlein, Brightness-normalized partial least squares regression for hyperspectral data, Journal of Quantitative Spectroscopy and Radiative Transfer 111 (12-13) (2010) 1947–1957.

[31] X. Fan, Y. Liu, J. Tao, Y. Weng, Soil salinity retrieval from advanced multi-spectral sensor with partial least square regression, Remote Sensing 7 (1) (2015) 488–511.

[32] K. Meacham-Hensold, C. M. Montes, J. Wu, K. Guan, P. Fu, E. A. Ainsworth, T. Pederson, C. E. Moore, K. L. Brown, C. Raines, et al., High-throughput field phenotyping using hyperspectral reflectance and partial least squares regression (plsr) reveals genetic modifications to photosynthetic capacity, Remote Sensing of Environment 231 (2019) 111–176.

[33] R. Fischer, N. Piatkowski, C. Pelletier, G. I. Webb, F. Petitjean, K. Morik, No cloud on the horizon: probabilistic gap filling in satellite image series, in: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2020, pp. 546–555.

[34] J. Chen, X. Zhu, J. E. Vogelmann, F. Gao, S. Jin, A simple and effective method for filling gaps in landsat etm+ slc-off images, Remote sensing of environment 115 (4) (2011) 1053–1064.

[35] X. Zhu, F. Gao, D. Liu, J. Chen, A modified neighborhood similar pixel interpolator approach for removing thick clouds in landsat images, IEEE Geoscience and Remote Sensing Letters 9 (3) (2011) 521–525.

[36] C. Zeng, H. Shen, L. Zhang, Recovering missing pixels for land-

sat etm+ slc-off imagery using multi-temporal regression analysis and a regularization method, Remote Sensing of Environment 131 (2013) 182–194.
[37] Q. Cheng, H. Shen, L. Zhang, Q. Yuan, C. Zeng, Cloud removal for remotely sensed images by similar pixel replacement guided with a spatio-temporal mrf model, ISPRS journal of photogrammetry and remote sensing 92 (2014) 54–68.
[38] F. Melgani, Contextual reconstruction of cloud-contaminated multitemporal multispectral images, IEEE Transactions on Geoscience and Remote Sensing 44 (2) (2006) 442–455.
[39] B. Chen, B. Huang, L. Chen, B. Xu, Spatially and temporally weighted regression: A novel method to produce continuous cloud-free landsat imagery, IEEE Transactions on Geoscience and Remote Sensing 55 (1) (2016) 27–37.
[40] Z. Zhu, C. E. Woodcock, C. Holden, Z. Yang, Generating synthetic landsat images based on all available landsat data: Predicting landsat surface reflectance at any given time, Remote Sensing of Environment 162 (2015) 67–83.
[41] W.-J. Zheng, X.-L. Zhao, Y.-B. Zheng, J. Lin, L. Zhuang, T.-Z. Huang, Spatial-spectral-temporal connective tensor network decomposition for thick cloud removal, ISPRS Journal of Photogrammetry and Remote Sensing 199 (2023) 182–194.
[42] P. Ebel, V. S. F. Garnot, M. Schmitt, J. D. Wegner, X. X. Zhu, Uncrtaints: Uncertainty quantification for cloud removal in optical satellite time series, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 2085–2095.
[43] Q. Zhang, Q. Yuan, J. Li, Z. Li, H. Shen, L. Zhang, Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning, ISPRS Journal of Photogrammetry and Remote Sensing 162 (2020) 148–160.
[44] Q. Zhang, Q. Yuan, Z. Li, F. Sun, L. Zhang, Combined deep prior with low-rank tensor svd for thick cloud removal in multitemporal images, ISPRS Journal of Photogrammetry and Remote Sensing 177 (2021) 161–173.
[45] X. Zhao, K. Jia, Cloud removal in remote sensing using sequential-based diffusion models, Remote Sensing 15 (11) (2023) 2861.
[46] X. Zou, K. Li, J. Xing, Y. Zhang, S. Wang, L. Jin, P. Tao, Diffcr: A fast conditional diffusion framework for cloud removal from optical satellite images, IEEE Transactions on Geoscience and Remote Sensing 62 (2024) 1–14.
[47] C. Stucker, V. S. F. Garnot, K. Schindler, U-TILISE: A sequence-to-sequence model for cloud removal in optical satellite time series, IEEE Transactions on Geoscience and Remote Sensing 61 (2023).
[48] R. Bellman, A markovian decision process, Journal of mathematics and mechanics (1957) 679–684.
[49] M. Schmitt, L. H. Hughes, C. Qiu, X. X. Zhu, Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion, arXiv preprint arXiv:1906.07789 (2019).
[50] M. Buchhorn, B. Smets, L. Bertels, M. Lesiv, N.-E. Tsendbazar, D. Masiliunas, L. Linlin, M. Herold, S. Fritz, Copernicus Global Land Service: Land Cover 100m: Collection 3: epoch 2019: Globe (Version V3.0.1), Zenono. DOI: 10.5281/zenodo.3939050 (2020).
[51] P. Ebel, Y. Xu, M. Schmitt, X. X. Zhu, SEN12MS-CR-TS: A Remote Sensing Data Set for Multi-modal Multi-temporal Cloud Removal, IEEE Transactions on Geoscience and Remote Sensing (2022).
[52] V. Sarukkai, A. Jain, B. Uzkent, S. Ermon, Cloud removal from satellite images using spatiotemporal generator networks, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1796–1805.
[53] V. S. F. Garnot, L. Landrieu, Panoptic segmentation of satellite image time series with convolutional temporal attention networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4872–4881.
[54] Google, Google Earth Engine, `https://earthengine.google.com/`, accessed: 16-03-2022 (2022).
[55] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, F. Hutter, Auto-sklearn: efficient and robust automated machine learning, in: Automated Machine Learning, Springer, Cham, 2019, pp. 113–134.
[56] J. Mockus, Bayesian approach to global optimization: theory and applications, Vol. 37, Springer Science & Business Media, 2012.
[57] M. Czerkawski, R. Atkinson, C. Michie, C. Tachtatzis, Satellitecloudgenerator: Controllable cloud and shadow synthesis for multi-spectral optical satellite images, Remote Sensing 15 (17) (2023). `doi:10.3390/rs15174138`.
URL `https://www.mdpi.com/2072-4292/15/17/4138`
[58] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (1989) 600–612.