Seeing Through Clouds in Satellite Images

Mingmin Zhao\textsuperscript{©}, Member, IEEE, Peder Olsen\textsuperscript{©}, Senior Member, IEEE, and Ranveer Chandra\textsuperscript{©}, Fellow, IEEE

Abstract—This article presents a neural network-based solution to recover pixels occluded by clouds in satellite images. We leverage radio frequency (RF) signals in the ultrahigh/superhigh-frequency band that penetrates clouds to help reconstruct the occluded regions in multispectral images. We introduce the first multimodal multitemporal method for cloud removal. Our model uses publicly available satellite observations and produces daily cloud-free images. Experimental results show that our system outperforms several baselines on multiple metrics. We also demonstrate use cases of our system in digital agriculture, flood monitoring, and wildfire detection.

Index Terms—Cloud detection, image inpainting, image reconstruction, remote sensing, sensor fusion, synthetic aperture radar (SAR).

I. INTRODUCTION

OBSERVING and monitoring Earth from space is crucial for tackling some of the greatest challenges of the 21st century, such as climate change, natural resources management, and disaster mitigation. Recent years have witnessed a surge of interest in Earth observation satellites that orbit around the planet and capture information about the Earth’s surface. The increased availability of satellite images with high resolution and short revisit time have enabled numerous applications in digital agriculture [1], [2], environmental monitoring [3], [4], [5], [6], humanitarian assistance [7], [8], transport and logistics [9], [10], and many more.

Satellites with optoelectronic sensors passively examine the Earth’s surface across visible and infrared bands to capture images. The resulting multispectral image is similar to an RGB image except that it has more channels corresponding to different wavelengths. However, one fundamental challenge in optical satellite images is occlusion due to clouds. Approximately 55\% of the Earth’s surface is covered in opaque clouds with an additional 20\% being obstructed by cirrus or thin clouds [11], [12], [13]. As a result, these cloud pixels are typically detected and discarded before further analysis. Cloud occlusions limit the timely detection of changes (e.g., crop emergence and flooding), thereby missing opportunities for early intervention. The problem is exacerbated by the intermittent and irregular availability of cloud-free pixels, which poses challenges for statistical analysis and machine learning on top of satellite images.

In this article, we aim to recover the pixels that are occluded by clouds. Unlike existing methods on cloud removal [14], [15], [16], [17], [18], [19], [20], [21], [22] or, more generally, image and video inpainting [23], [24], [25], [26], [27], [28], [29], [30] that generates visually plausible inpainting results based on the context, our goal is to recover optical observations accurately. Our main idea is to use a different modality, i.e., radio frequency (RF) signals, to help recover the optical information. While clouds mostly block visible and infrared wavelengths, RF signals in the ultrahigh/superhigh frequencies penetrate clouds. Furthermore, these signals reflect off the Earth’s surface. Using a synthetic aperture radar (SAR) technique, one can generate a radar image that captures the surface’s radio reflectance even in the presence of clouds at night. Existing work has used radar images to detect changes in radio reflectance due to metallic objects [31], [32], water bodies [33], [34], and geometric transformations of landscape (e.g., constructions and Earthquakes) [35], [36]. More recent work has tried to translate SAR images into multispectral images [37], [38]. However, these methods do not consider the temporal history of multispectral images and have limited accuracy (see Table I).

We introduce SpaceEye, a neural-network-based Earth observation system that achieves the best of both optical and RF modalities, i.e., it captures rich information in multispectral bands despite clouds and lighting conditions. To achieve this, we introduce the first cloud removal model that makes use of both multimodal and multitemporal satellite observations. It leverages multitemporal satellite observations in multispectral and RF bands\textsuperscript{1} to recover daily cloud-free multispectral images.

Fig. 1 shows an example output of our system tracking the floods in a farm located in Carnation, WA, USA. It shows the sequences of multispectral and SAR images captured during this period. As can be seen, most of the multispectral images capture clouds instead of the farm underneath them. While SAR images do not capture as rich information as multispectral images could, they provide consistent observation of RF reflectivity on the ground. Our system generates daily cloud-free multispectral predictions, as shown in the bottom row. As can be seen, it captures the extent of the standing water on February 7, despite the fact that the multispectral image on the same day was covered in clouds. It also shows how the flood water gradually drained off after mid-February.

There are several challenges in designing and training our system. First, effectively fusing multispectral images and

\textsuperscript{1}SpaceEye uses publicly available data from the European Space Agency (ESA) governed by the “Legal Notice on the use of Copernicus Data and Service Information.” ESA provides up-to-date multispectral and RF measurements from the different Sentinel missions [39], [40].

Manuscript received 27 June 2022; revised 15 November 2022; accepted 11 December 2022. Date of publication 25 January 2023; date of current version 8 September 2023. This work was supported by Microsoft Corporation.
(Corresponding author: Mingmin Zhao.)

Mingmin Zhao is with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: mingminz@seas.upenn.edu).
Peder Olsen and Ranveer Chandra are with Microsoft Research, Microsoft Corporation, Redmond, WA 98052 USA.
Digital Object Identifier 10.1109/TGRS.2023.3239592

1558-0644 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Fig. 1. SpaceEye in action: it takes as input multispectral images (top row) and SAR images (middle row), and produces cloud-free predictions (bottom row). The scene shows a flooded area in Carnation in 2020. Note the infrequent availability of cloud-free multispectral images during the flooding event. Our model reconstructs the multispectral images covered by clouds, capturing the extent of standing water on February 7 and how the flood drained off after mid-February.

RF signals is challenging. These multispectral images and RF signals are captured by different satellites on separate orbits. Yet, the clouds in the multispectral images have a nonpredictable and free-form distribution. Second, there are no ground-truth cloud-free images that can be used during training. This poses challenges for supervised training and limits adversarial training, which can help ensure the fidelity of the predictions.

In this article, we tackle the above challenges as follows. We use a technique called SAR to convert RF time signals into 2-D images and align them with the multispectral images. We design a multimodal attention mechanism to fuse this multimodal data. Specifically, we use RF information to decide where each pixel should be attended to when reconstructing the multispectral images. We also propose a new adversarial training setting that only requires partial observations of examples in the target distribution. This modified adversarial training setting allows us to ensure the fidelity of cloud-free predictions in the absence of ground-truth cloud-free images.

We train on a dataset containing both the multispectral and RF observations spanning three years and covering 29 regions of size $100 \times 100$ km$^2$ across the U.S. and Europe. We test it in various areas from the U.S., Europe, and Africa. Experiments show that SpaceEye achieves a peak signal-to-noise ratio (PSNR) of 28.9, which significantly outperforms baselines. We further conduct several case studies to demonstrate the use of SpaceEye to agriculture, wildfire monitoring, and storm damage assessment.

II. RELATED WORK

A. Cloud Removal in Satellite Images

Cloud removal starts by detecting cloud pixels [41], [42], [43], [44], [45], [46] and then reconstructs these pixels based on the remaining ones. The effectiveness of existing cloud removal methods is tightly coupled with the amount of input information. Existing work generally falls into two categories: single-image and multitemporal methods. Single-image methods take a single satellite image as input. Various image inpainting algorithms have been used for this task, including sparse dictionary learning [47], [48] and neural-network-based approaches [15], [16], [17], [18], [19]. These algorithms generate visually realistic and semantically correct images. However, inpainting-based methods have limited accuracy for large missing regions, which is common due to the prevalence of clouds. Multitemporal methods [20], [21], [22] use multiple satellite images from nearby days to produce a single cloud-free image, e.g., patch-based methods [49], [50], [51] and tensor factorization approaches [52], [53], [54], [55]. Several systems [56], [57], [58], [59] further fuse multispectral images sequences from multiple satellites to increase the number of images in any given time range. The performance of these systems is much improved compared to the single-image methods. However, these systems assume little change between cloud-free observations and cannot detect abrupt changes. RF information has also been used as auxiliary information to help remove clouds [37], [38], [60]. These methods use the SEN12MS dataset [61], which provides pairs of multispectral and RF images, but do not consider the temporal sequence of satellite images. Collectively, our work distinguishes itself from previous methods by straddling both the multimodal and multitemporal domains, i.e., it utilizes sequences of multispectral and SAR images for cloud removal. Experimental results show the advantages of our method (see Section VI-B).

B. Image and Video Inpainting

Both traditional diffusion or patch-based methods [62], [63], [64], [65], [66] and learning-based methods [24], [25], [26], [27], [28], [29], [30] have been used for image and video
inpainting. We build on this literature and adopt a learning-based approach but differ in that we use multimodal data and aim to recover the actual observations occluded by clouds. While the attention mechanism has been widely used in the inpainting models [25], [26], we used auxiliary RF information from the missing region to guide the attention instead of using spatial context from nearby pixels.

C. Learning With Multiple Modalities

Our work is related to cross-modal and multimodal learning that leverages complementary information across modalities [67], [68], [69], [70], [71], [72], [73]. Multimodal learning of images and RF have been explored to sense people through walls [74], [75], [76]. While self-supervised and unsupervised methods were typically used to guide the representation learning, we use multimodal data to recover missing data and focus on the task of cloud removal.

III. SYNTHETIC APERTURE RADAR IMAGING

SAR is a remote sensing method that has attracted much interest. Aircraft or spacecraft equipped with SAR actively transmits pulses of radio waves to “illuminate” the target region and record the waves reflected after interacting with the Earth. The recorded SAR data are very different from optical images, which can be interpreted similar to a photograph. Radio waves are time signals that are responsive to surface characteristics, such as structure and moisture. Radar sensors usually utilize longer wavelengths at the centimeter-to-meter scale, which penetrates clouds.

Radar imaging algorithms transform the recorded radio data into 2-D images. The spatial resolution of the resulting radar image is directly related to the ratio of the antenna size to the signal wavelength. However, getting a spatial resolution of 10 m from a satellite operating at a wavelength of about 5 cm requires a radar sensor about 4250 m long, which is not practical. SAR is a clever workaround that leverages the synthetic aperture. Specifically, SAR uses the satellite’s motion over the target region, as shown in Fig. 2. A sequence of acquisitions from a smaller radar sensor is combined to simulate a much larger antenna, thus providing higher resolution SAR images [77].

We use Sentinel-1 SAR products made publicly available by the ESA. We process the raw dual-polarization Level-1 products with the SAR imaging algorithm and other processing steps, including orbit correction, noise removal, calibration, and terrain correction. Please refer to the Supplementary Material for a detailed description of the SAR processing.

IV. METHOD

We propose a neural network framework that uses multispectral and RF satellite images to reconstruct cloud-free multispectral images. Fig. 3 illustrates the model architecture of SpaceEye. It takes two streams of satellite images (i.e., multispectral and SAR) as input and uses a coarse-to-fine two-stage network to predict cloud-free images. In the first stage, a coarse network encodes input multispectral and SAR images and learns to recover the missing pixels via masked attention (see Section IV-A). In the second stage, a refinement network further improves the predictions through a pair of encoder and decoder with skip connections (see Section IV-B). To ensure the fidelity of the predictions, we propose a new adversarial training setting that trains the discriminator with partial observations, i.e., it is trained with satellite images that are often cloudy (see Section IV-C). We describe how we train SpaceEye (see Section IV-D) followed by a matrix completion variant of it (see Section IV-E).

A. Multimodal Fusion With Attention

The first stage of our model takes the input sequence of multispectral and SAR images, and extracts an encoding for each space–time location. Our model considers the spatial and temporal contexts to recover the multispectral encoding for space–time locations that are occluded by clouds. Specifically, it uses an attention mechanism to effectively borrow information from other possibly distant space–time locations. It uses SAR encodings that are available for all space–time locations to guide the attention mechanism. This network, finally, uses a decoder to transform the recovered encoding back to the multispectral space. We describe each component of the coarse network in detail.

1) Encoders: The coarse network uses two encoders for multispectral and SAR images. Each encoder has nine convolution layers with BatchNorm [78] and LeakyReLU [79]. It encodes each image from the time sequence separately with $3 \times 3$ convolution kernels. The encoders reduce the spatial resolution to $1/8$ while maintaining the temporal resolution.

2) Multimodal Attention: Our model uses attention mechanism [80], [81] to reconstruct the space–time locations occluded by clouds. A key component of the attention mechanism is the key and query vectors that allow for the computation of attention scores. We designed a multimodal attention mechanism that computes key and query vectors based on the SAR encodings, which are always available despite clouds. Moreover, we use masked attention to make sure that the model will only learn to attend to cloud-free space–time locations. The mask that we use here is the cloud mask from the S2Cloudless algorithm [46].

Precisely, our neural attention model computes output $\mathbf{h}$ from SAR encoding $\mathbf{r}$, multispectral encoding $\mathbf{o}$, and cloud mask $\mathbf{m}$ (1 for cloud-free pixels and 0 otherwise) as follows:

$$
\mathbf{h}_i = \frac{1}{C(\mathbf{r}, \mathbf{m})} \sum_{\mathbf{j}} a(\mathbf{r}_i, \mathbf{r}_j, \mathbf{m}_j) \cdot V(\mathbf{o}_j)
$$

(1)
where \( i \) is the index of an output position in space–time and \( j \) is the index that enumerates all possible positions. The value vector \( V(\cdot) \) is computed using multispectral encodings\(^2\) \( o \). The attention scores \( a(\cdot) \) are normalized by a factor \( C(r, m) \).

We use a dot product followed by a softmax operation to compute the attention scores as

\[
a(r_i, r_j, m_j) = e^{K(r_j)^T Q(r_i) \cdot m_j} \tag{2}
\]

where the key \( K(\cdot) \) and the query \( Q(\cdot) \) are computed using SAR encodings \( r \). Note that the dot product score is multiplied by the cloud mask, which forces the attention module to ignore space–time positions with clouds (i.e., \( m_j = 0 \)).

3) **Decoder:** The decoder has three upsampling layers to recover the original spatial resolution. Each upsampling layer is followed by two \( 3 \times 3 \) convolution layers.

**B. Refinement Network**

We follow Yu et al. [25] to design a coarse-to-fine network architecture that progressively improves the quality of the predictions. We use a variant of the U-Net architecture [82] with 3-D spatial–temporal convolutions. The **encoder** has seven downsampling blocks with \( 3 \times 4 \times 4 \) convolution kernels, BatchNorm, and LeakyReLU. The first three blocks have a stride of \( 2 \times 2 \times 2 \), while the rest uses \( 1 \times 2 \times 2 \). Each block doubles the number of channels. The **decoder** has seven upsampling blocks with transposed convolution. It also uses skip connection from the intermediate representations of the encoder.

**C. Adversarial Training With Partial Observations**

We would like to ensure the fidelity of our model’s prediction, i.e., we want the predicted cloud-free multispectral images and their temporal evolution to be realistic. An effective way to ensure fidelity is to use adversarial training [25], [83], [84], [85]. At the equilibrium state of adversarial training, the generated distribution will converge to the target distribution (i.e., distribution of positive examples). However, typical adversarial training would not work in our case since the ground-truth multispectral images contain clouds that cannot be used during training.

To address this issue, we introduce a new adversarial training setting that can learn with partial observations of the target examples. In this setting, the discriminator no longer has full access to either positive nor negative examples. Instead, it will only have access to a randomly masked version of the examples. As illustrated in Fig. 4, the discriminator only has access to the cloud-free pixels for positive examples, i.e., they will be masked by the cloud detection results. On the other hand, the negative examples generated by our decoder will be masked with a randomly generated cloud mask before sending to the discriminator. Mathematically, (3) shows the value function for the vanilla adversarial training, and (4) shows the value function for the partial-observation-based adversarial training. \( p_{\text{data}} \) and \( p_{\text{model}} \) denote the distribution of real multispectral images and the distribution of the generated cloud-free multispectral images, respectively, \( m \) denotes the cloud mask, and \( D \) is the discriminator.

\[
\begin{align*}
&\mathbb{E}_{x \sim p_{\text{data}}} \log D(x) + \mathbb{E}_{x \sim p_{\text{model}}} \log (1 - D(x)) \tag{3} \\
&\mathbb{E}_{x, m \sim p_{\text{data}}} \log D(x \cdot m) + \mathbb{E}_{x \sim p_{\text{model}}, m} \log (1 - D(x \cdot m)). \tag{4}
\end{align*}
\]
Consider the equilibrium states of adversarial training using the objectives defined in (3) and (4). The vanilla adversarial training reaches equilibrium when \( p_{\text{model}} \) converges to \( p_{\text{data}} \). This is, however, not ideal for the task of cloud removal since \( p_{\text{data}} \) is not cloud-free. On the other hand, the equilibrium state of training with (4) ensures that randomly sampled patches of the prediction \( p_{\text{model}} \) converge to the cloud-free patches from \( p_{\text{data}} \).

We use partial convolution [27] as the basic building blocks for our discriminator. It ensures that the output of the convolution operation is properly masked and renormalized to be conditioned on valid pixels. Our discriminator uses five 3-D partial convolutions with \( 3 \times 4 \times 4 \) kernels and \( 1 \times 2 \times 2 \) strides.

D. Training of SpaceEye

1) Training With Synthetic Clouds: We use synthetic clouds to train the coarse network and the refinement network. Specifically, we mask out (i.e., set to zeros) additional portions of the multispectral images that are cloud-free and use them to provide supervision during training. We generate the synthetic cloud mask by randomly sampling another cloud mask from the training set. We update the cloud mask to the union of the original cloud mask and the newly sampled cloud mask. Note that we do not need to render synthetic clouds in the original image since all cloudy pixels are set to zeros before being passed to the model. This training strategy is similar in spirit to the training strategy commonly used in image inpainting models [25], [26], [27].

2) Loss Functions: We use \( L_1 \) loss for the output of the coarse network and the refinement network, comparing them with the held-out synthetic cloud pixels. We train the discriminator using the loss defined in (4). The corresponding adversarial loss for the generator is computed by flipping the labels. We add the adversarial loss and the \( L_1 \) losses with a weight of 20.0 for \( L_1 \) losses.

3) Training Details: We randomly crop multispectral and SAR images to the size of \( 256 \times 256 \) and use sequences of length 48 spanning 48 days. We implemented our model in PyTorch. It is trained with the Adam [86] optimizer for 60,000 iterations. We use a batch size of 4 on two GPUs and an initial learning rate of \( 2e-5 \) with ten times decay every 15,000 iterations. The training takes two days.

E. Matrix Completion Variant of SpaceEye

We provide a matrix completion variant of SpaceEye, which we refer to as SpaceEye-MC. Intuitively, viewing a time series of satellite images as a big matrix with clouds as missing entries in it, we can recover pixels covered by clouds through matrix completion. We format the matrix such that the first dimension of our matrix is spatial locations, and the second dimension is time \( \times \) channels (both multispectral and SAR channels). Formatting the matrix this way allows for a low-rank decomposition representing pixels as mixtures of land-types, each of which has its land evolution signature.

Let \( C_1 \times T \times H \times W \) and \( C_2 \times T \times H \times W \) denote the size of the temporal sequence of multispectral and SAR observations. We concatenate and reshape the tensors above to form the observation matrix \( Y \in \mathbb{R}^{(C_1+C_2)T \times HW} \). We also use a matrix \( M \) of the same size to indicate the availability of the entries in the observation matrix \( Y \). Our goal is to find \( X \), a low-rank approximation of \( Y \), while maintaining the smoothness of observations over time. Let \( X_t \in \mathbb{R}^{(C_1+C_2) \times HW} \) denote the elementwise time slices of \( X \) representing the observations at time \( t \). We find \( X \) by solving the following rank-constrained optimization:

\[
\min_{\text{rank}(X) \leq r} \| M \circ (X - Y) \|_F^2 + \alpha \sum_{t=1}^{T-1} \| X_{t+1} - X_t \|_F^2 \quad (5)
\]

where \( \alpha \) is a damping factor and \( \circ \) is the elementwise multiplication operator. Solving it without the rank constraint decouples channels and becomes damped interpolation. Both optimization problems can be solved efficiently on a GPU. We outline the details in the Supplementary Material.

V. DATASET

We extracted and aligned radar and multispectral images using data from ESA’s Sentinel-1 and Sentinel-2 missions. The data was downloaded using sentinelsat licensed under GPLv3+.

Sentinel-1 is a constellation of two satellites that have been operating since April 2016 with a joint location revisit rate of three days at the equator and two days in Europe. Sentinel-1 carries a single C-band SAR instrument operating centering 5.405 GHz. For our experiments, we used data from the interferometric wide (IW) swath mode that covers 250 km with dual-polarization VV-VH.

Sentinel-2 is a system of two satellites with a joint revisit time of five days that have been in operation since March 2017. The satellites carry a multispectral instrument (MSI) with 13 spectral channels in the visible, near-infrared (NIR), short-wave infrared (SWIR) spectral range, and so on. We use Level 1 top-of-the-atmosphere reflectance products in our experiments.

A. Data Preparation

The data preparation involved handling edge effects, merging images on the same absolute orbit, and marking missing pixels. The Sentinel-1 radar processing additionally involves thermal noise removal, calibration, terrain correction, and so on. The radar processing was done using snap licensed under GPLv3, and the details are described in the Supplementary Material. We scale multispectral images to be in the \([0, 1]\) interval and SAR images to be in \([-1, 1]\).

B. Training and Test Data

For the training data, we used 32,000 randomly selected spatiotemporal regions spanning 48 days and images of size \( 256 \times 256 \) from 29 tiles. The 29 tiles were taken from Europe, Iowa, and Washington between January 2018 and August 2020. For the validation test, we used 30 spatiotemporal regions of the same size from three other tiles. Although this only amounts to 0.1% of the amount of training data,
the validation data are only used to decide the final epoch for training. For testing, we used 1000 randomly selected spatiotemporal regions spanning 48 days and images of size 448 × 448 from nonoverlapping tiles taken from Washington, Iowa, Europe, Rwanda, and Australia. This amounts to 10% of the amount of training data, but the geographical diversity is larger.

Fig. 5 shows the probability distribution of the cloud cover ratio for the test data, i.e., 48-day sequences of multispectral images. There is not a single sequence that is cloud-free, and in fact, a very low fraction of the data has a cloud cover below 50%.

VI. EVALUATION

In this section, we show qualitative and quantitative results of SpaceEye, compare it with several baseline algorithms, and demonstrate example applications of it.

A. Qualitative Evaluation of SpaceEye

Fig. 6 shows the results of SpaceEye on the test data. The first three rows show the SAR image, multispectral image, and our prediction of the same day. Note that time sequences of multispectral and SAR images are used as input. To get a sense of what the rest of the images might look like and help understand the prediction, we visualize the nearest cloud-free or the least cloudy multispectral image before and after inference time. Comparing the last three rows, our model generates accurate and semantically correct cloud-free images for various land types, including farmlands, hills, villages, and so on. Moreover, our model accurately tracks the changes on the ground. For example, Fig. 6(a) shows how it captures harvesting activities: while no fields were harvested in the previously available image and most of them are harvested in the next available image, our prediction shows the field that was first harvested. Similarly, our model captures the changes due to growing plants and agriculture products in Fig. 6(b)–(j), as well as dissolving snow in Fig. 6(e) and (g). Fig. 6(f) shows another example of how our model deals with the cirrus cloud.

B. Quantitative Evaluation of SpaceEye

1) Baselines: We consider two multitemporal baselines, namely, SpaceEye-MC and damped interpolation (see Section IV-E). We use τ = 35 and α = 3 for SpaceEye-MC and α = 0.5 for damped interpolation. We also consider three single-image-based baselines: a Cycle-GAN-based method that translates an SAR image into a multispectral image [37] (SAR2optical), an image inpainting model [26] retrained with multispectral images (DeepFill v2), and an SAR-optical fusion [38] method that uses an SAR image together with a multispectral image to remove clouds (DSen2-CR).

2) Evaluation Metrics: To evaluate the cloud removal performance, we used the concept of synthetic clouds where we hold out (i.e., set to zeros) parts of the clear images. We report reconstruction errors on synthetic clouds (syn), as well as reconstruction errors on synthetic clouds and cloud-free pixels (all). The metrics that we report are PSNR, the mean absolute error (MAE), the Pearson correlation coefficient (R²), and the multiscale structural similarity index measure (MS-SSIM) [87]. The MS-SSIM metrics are interesting in a simulated context but are less meaningful when the ground-truth images contain clouds.

3) Results: Table I shows the performance of SpaceEye and baselines across all ten spectral bands. As shown in the tables, SpaceEye significantly outperforms all the baselines. We can also observe that all three multitemporal methods perform better than the three methods that do not use temporal history. Among these single-image methods, DSen2-CR [38], which uses both optical and RF inputs, has the best performance. Note that both DSen2-CR and DeepFill v2 leave cloud-free pixels unchanged, thus inflating the results for the all category. The other methods modify cloud-free pixels to attempt a natural transition (in space and time) between cloud-free and inpainted regions.

Table II breaks down the PSNR for each individual spectral band. The table shows that the PSNR is lower for the red, vegetation red-edge bands, and NIR bands, as these bands have a higher complexity affected by many plant traits (e.g., leaf chlorophyll and nitrogen content).

4) Performance Versus Cloud Cover Ratio: We look at the performance of each algorithm as a function of the cloud cover ratio. We grouped 1000 test entries according to the cloud cover ratio and computed their MAE. Fig. 7 plots the median for each method for cloud-fractions ranging from 0.3 to 0.95. The shaded area additionally shows the 25%–75% quantiles of the error. For multitemporal methods, the median MAE and its variation increase with the cloud cover ratio. The single-scene methods using RF, on the other hand, are less affected by the increased cloud cover ratio. Interestingly, DSen2-CR outperforms both SpaceEye-MC and damped interpolation when the cloud ratio is very large, showing the advantages of neural network training over optimization-based methods. Among all the methods, SpaceEye is robust in the presence of clouds with a smaller shaded area and provides the best results uniformly.

C. Case Studies of SpaceEye Applications

We conduct case studies to demonstrate the applications of SpaceEye.

3We used a custom implementation of MS-SSIM that appropriately ignores masked out regions of the image.
Fig. 6. SpaceEye results on the test set. The first three rows show the SAR image, multispectral image, and SpaceEye’s prediction of the same day. The last two rows labeled show the nearest cloud-free or the least cloudy multispectral image before and after. The labels on the images show the time relative to the inference time. Columns (a)–(j) are sampled data from the test set.

### TABLE I

| Methods                  | input modalities | multi-temporal? | all  | syn       |
|--------------------------|------------------|-----------------|------|-----------|
|                          |                  |                 | PSNR | MS-SSIM   | MAE  | R²     | PSNR | MS-SSIM | MAE | R²     |
| SpaceEye (Ours)          | optical & RF     | ✓               | 28.36| 0.966 | 0.014 | 0.955 | 23.30| 0.902 | 0.031 | 0.847 |
| SpaceEye-MC              | optical & RF     | ✓               | 26.12| 0.951 | 0.019 | 0.925 | 21.37| 0.867 | 0.040 | 0.762 |
| Damped Interp.           | optical          | ✓               | 26.94| 0.965 | 0.014 | 0.938 | 21.22| 0.866 | 0.041 | 0.752 |
| DSen2-CR [38]            | optical & RF     | x               | 26.19| 0.923 | 0.019 | 0.926 | 20.54| 0.764 | 0.053 | 0.684 |
| DeepFill v2 [26]         | optical          | x               | 21.92| 0.914 | 0.029 | 0.819 | 16.05| 0.647 | 0.111 | 0.339 |
| SAR2optical [37]         | RF               | x               | 19.52| 0.752 | 0.056 | 0.597 | 19.99| 0.754 | 0.054 | 0.623 |

### TABLE II

PSNR on Individual Spectral Band for Different Methods on All Synthetic Clouds and Cloud-Free Pixels (VRE = Vegetation Red Edge; NIR = Near-Infrared; NNIR = Narrowband NIR; and SWIR = Short-Wave Infrared)

| Methods                  | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 8A | 11 | 12 |    |
|--------------------------|---|---|---|---|---|---|---|---|----|----|----|
| SpaceEye (Ours)          | 29.10 | 29.17 | 28.31 | 28.49 | 27.76 | 27.11 | 27.04 | 26.83 | 30.36 | 31.71 |
| SpaceEye-MC              | 26.57 | 27.04 | 25.95 | 26.25 | 25.42 | 24.91 | 24.80 | 24.58 | 28.34 | 30.10 |
| Damped Interp.           | 27.17 | 27.65 | 26.93 | 26.90 | 26.01 | 25.43 | 25.66 | 25.24 | 30.31 | 33.03 |
| DSen2-CR [38]            | 26.83 | 27.23 | 26.10 | 26.55 | 25.85 | 24.59 | 24.64 | 24.32 | 28.56 | 30.91 |
| DeepFill v2 [26]         | 23.55 | 24.62 | 23.38 | 23.57 | 21.09 | 19.46 | 19.26 | 19.66 | 24.32 | 27.26 |
| SAR2optical [37]         | 19.27 | 19.76 | 18.89 | 19.11 | 18.90 | 18.46 | 18.59 | 18.50 | 22.47 | 24.77 |

1) **Case Study I (Digital Agriculture):** In Fig. 8, we show an application of satellite imaging to agriculture. We calculate the normalized difference vegetation index (NDVI) used as a qualitative indication of crop health and vitality, and the normalized difference water index (NDWI) used to monitor leaf water content. To calculate NDVI and NDWI, the NIR and SWIR bands are used. These spectral bands are also produced by SpaceEye. In the images, we can see in-field variations of these indices. Normally, NDVI and NDWI would only be calculated on a cloud-free satellite image, but SpaceEye can provide them even on cloudy days, thereby enabling timely interventions for growers, such as for irrigation, pesticides, nutrients, and other farm management practices.

2) **Case Study II (Storm Damage Assessment):** On August 9, 2020, a derecho swept through the state of Iowa and caused billions of dollars of damage. In Fig. 9, we apply SpaceEye to detect changes due to the storm. By using SpaceEye, we do not need to wait for a completely cloud-free
Fig. 7. MAE for all cloud-free pixels as a function of the percentage of cloudy pixels. The shaded area shows the 25%–75% quantiles, and the solid line shows the median for each method.

Fig. 8. SpaceEye inference in Quincy, WA, USA. The plots show the RGB prediction along with the NDVI and NDWI.

Fig. 9. Change detection applied to an image from Luther, IA, USA, before and after a derecho storm on August 9, 2020.

Fig. 10. Wildfire detection with normalized burn ratio (NBR).

multispectral image to do the damage assessment. This can trigger a timely response from insurers and the government.

3) Case Study III (Wildfire Monitoring): Fig. 10 shows a multispectral image of the wildfire in Bridgeport, WA, USA, from September 2020 along with the change in the normalized burn index comparing with the previous year using SpaceEye. The change in the normalized burn index can be used to monitor the burned area and the severity of the burn.

D. Discussions and Limitations

The system is limited in that it cannot handle extremely cloudy situations nor gracefully recover from cloud-detection errors. When there is a lack of cloud-free patches, the attention mechanism cannot accurately reconstruct the cloudy data. Moreover, since the attention mechanism’s memory requirement is quadratic in the number of pixels, the size of the window cannot be significantly enlarged. However, a future improvement could be to dynamically change the window in time and space according to the location near-est cloud-free imagery. Other possible future improvements could be to aggressively factor the attention. Also, due to the attention mechanism, failures in the cloud detector can propagate beyond its original location. This is an effect that we have observed for large water bodies where the cloud detector is less accurate. Finally, future research could also incorporate landcover classes to improve the reconstruction accuracy [88], [89].

VII. CONCLUSION

This article presents a new technology to see through clouds in satellite imagery. It recovers daily cloud-free multispectral...
images with high accuracy by learning from multimodal and multitemporal satellite observations, leading to 2-dB improvement compared to several baseline algorithms.

SpaceEye builds on a growing body of work in multimodal sensing and learning. It shows that RF signals, a sensing modality with intrinsically different properties than visible light, can augment vision systems with powerful capabilities. This also marks an important step in Earth observation and its applications. We use computer vision to solve a fundamental challenge in remote sensing, which significantly improves the quality of satellite images. SpaceEye also enables a suite of new applications, including timely interventions for agriculture, disaster assessment, and wildfire response.

We believe that SpaceEye opens up exciting opportunities for new applications that could be built on top of these daily cloud-free satellite images. Moving forward, we are partnering with the government, insurance, and agriculture companies to develop new applications of SpaceEye, including forest health monitoring, damage assessment from extreme weather events, wildfire prediction, precision agriculture, and measurement of greenhouse gas from farms.

**APPENDIX**

This appendix is organized as follows. Section A shows SpaceEye predictions for a farm in Washington state for the entire growing season. Sections B and D describe the processing needed for efficient training and inference with SpaceEye for the multispectral and the RF data, respectively. Section C discusses the methods used to detect clouds and cloud shadows. Finally, Section E derives and describes the damped interpolation method, and Section F derives the SpaceEye-MC method.

### A. Qualitative Comparison

In this section, we show all the available Sentinel-2 images for a period from April 1, 2019, until September 15, 2019, for a single farm plot in Washington state in Fig. 11. A total of six Sentinel-2 images are available for every 15 days, but many are covered in clouds, and we have masked out pixels not belonging to the farm. Fig. 12 shows daily predictions using SpaceEye for the same time period. Such a cloud-free day-by-day representation greatly simplifies application development as the developer no longer needs to worry about data availability and handling of clouds. It is for example a trivial operation to generate the average NDVI and NDWI curves for the farm as in Fig. 13. NDWI values below 0 are a good indicator that there is very little water content in the crop. Consequently, the NDWI values could have been used as a guide to decide on when to harvest.

---

Fig. 11. Sentinel-2 images for a hilly farm for the 2019 growing season near Pullman, WA, USA. Only pixels belonging to the field are shown.

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 12. VIR predictions with SpaceEye for a hilly farm for the 2019 growing season near Pullman. Only pixels belonging to the field are shown.

Fig. 13. Daily NDVI and NDWI predictions for a hilly farm for the 2019 growing season near Pullman.

B. Multispectral Data Processing

We used multispectral data from ESA’s Sentinel-2 [90] for multispectral, which provides open access as governed by the “Legal Notice on the use of Copernicus Data and Service Information.” Sentinel-2 is a constellation of the twin satellites, Sentinel-2A and Sentinel-2B, launched into orbit in June 2015 and March 2017, respectively. The satellites are in a ten-day orbital cycle with a joint repeat cycle of five days. Each pass images a swath width of 290 km, and the temporal coverage increases for locations inside overlapping swaths. These satellites have a joint revisit time of five days and systematically image land surfaces between 56° S and 84° N.

The Sentinel-2 products are provided in tiles aligned with the military MGRS zones. Each tile or granule covers roughly 100 × 100 km of land and is projected onto one of 60 Universal Transverse Mercator (UTM) coordinate systems. The multispectral data available for Sentinel-2 contain 13 spectral bands with resolutions of 10 m × 10 m for the three visible (red–green–blue) and NIR bands, 20 m × 20 m for the three vegetation red edge (VRE), narrow NIR (NNIR), and the two SWIR bands, and 60 m × 60 m for the aerosol, water vapor, and cirrus bands. The resolution mismatch and lack of cloud shadow mask and other artifacts, such as image artifacts on edge and a cloud mask not ideal for SpaceEye, led us to process the original data available from Sentinel-2 into an analysis-ready format for training and inference.

1) Data Access: We used the python sentinelsat module [91] for programmatic access to the Copernicus Open Access Hub [39]. This tool was used for data not older than one year at the time of access. It can also be used for historical data, but only one product can be requested every 30 min. Alternatively, historical data can also be purchased from approved Copernicus vendors [92].

2) Image Fusion: The Sentinel-2 products provided do not always cover an entire tile (a.k.a. granule), and sometimes, several images will cover different sections of the granule.
When these images were temporally close (seconds apart) and on the same absolute orbit, we combined the images using [93]. Generally speaking, we see no apparent artifacts in the merged image for cloud-free pixels, but the clouds may have shifted between the merged images.

In addition, we also merged all images in a window for the time step requested by the neural network (one or two days). For a time step of one day, there were no additional image merges, while, for time steps larger than one day, we selected the first image with the most cloud-free pixels. The remaining images were either discarded or shifted to an image-free neighboring time step. For regions with extremely high cloud densities, where there are no cloud-free pixels in a 48-day period, we utilized the time step of two days to process 96 days at a time.4

3) Stacking and Resampling: Sentinel-2 products contain 13 spectral bands. Four bands (red, green, blue, and NIR) are provided at a 10 m × 10 m pixel resolution, and six bands (three VRE bands, NNIR, and two SWIR) are provided at 20 m × 20 m pixel resolution, while the last three bands (aerosol, water vapor, and cirrus) are at a 60 m × 60 m pixel resolution. In order for the training algorithms to be efficient, we resampled all bands using bicubic interpolation to a common 10 m × 10 m pixel resolution and stacked them in a single GEOTIFF file. The GEOTIFF used the default block size of 256 to allow for efficient retrieval of rectangular subsections of the full image. This format is similar to the Cloud Optimized GEOTIFF (COG) standard [94] but lacks the overview capability. We plan to comply with the COG standard in the future.

4) Missing Data: In tiles that intersect the edge of an individual data strip, there will be missing pixels. We treat missing pixels as if they were clouds. Specifically, we created a “cloud mask” that combined cloud shadows, clouds, and missing pixels into a single mask.

5) Edge Effects: Due to the particular geometrical layout of the focal plane, each spectral band of the MSI observes the ground surface at different times [95]. In particular, the extent captured for each spectral band differs, yielding an undesirable rainbow-like effect on the edge of the images. We removed these regions by considering all pixels with a nodata value in any band as a missing pixel to be masked out.

C. Detecting Clouds in Satellite Images

Clouds interfere with atmospheric correction algorithms, frequently occlude the surface, and add noise to surface observations. Thus, one of the first steps in processing satellite images is a cloud segmentation operation. Cloud shadows are another source of noise, and cloud shadow segmentation can additionally utilize the cloud locations, the solar illumination angle, and the satellite viewing angle.

Cloud detection and cloud-shadow detection are well-studied problems [41], [42], [43], [44]. Cloud detection can be divided into pixelwise, single-scene, and multitemporal algorithms. Pixelwise algorithms are threshold-based and use the entire available spectrum as in the case of Function of Mask (Fmask) [45], now in its fourth version [96]. Fmask is widely used and additionally provides a cloud-shadow mask as well. The cloud detection model used in this article was based on the S2Cloudless algorithm [46]. S2Cloudless is another pixelwise thresholding that uses the LightGBM decision tree model [97].

Single-scene algorithms can utilize the spatial context through deep learning approaches. Deep cloud detection largely relies on the U-net architecture and has been shown to outperform the Fmask algorithm [44], [98], [99]. The deep learning approach requires a large annotated dataset. Such a dataset is, however, not publicly available for Sentinel-2. Mohajerani et al. [100] proposed using a GAN to artificially generate realistic cloudy satellite images that could possibly be used to train such a model.

The most accurate algorithms are in the multitemporal context. For satellites such as Landsat, Sentinel, and MODIS with predictable revisit rates, this is the most accurate algorithm class. The algorithms are computationally more complex and costly but offer the highest performance. The state-of-the-art performance for Sentinel-2 is the MAJA algorithm [101]. Multitemporal cloud detection has also been done for the Pléiades satellite [102], where a less extensive temporal context is available.

1) Cloud Annotation: Sentinel-2 is a relatively new product compared to MODIS and Landsat, so there are few open datasets available for cloud and cloud shadow annotations. One available source was the Hollstein dataset [103]. It contained a list of pixels with one of six classes annotated (cloud, cloud shadow, clear, cirrus, snow, and water). Without the spatial and temporal context available, this meant that we were constrained to the pixelwise algorithms by using the Hollstein dataset.

2) Cloud Detection: The Hollstein dataset was used in building a pixelwise LightGBM [97] decision-tree-based model called S2Cloudless [46] for detecting clouds. We used the very efficient and lightweight S2Cloudless algorithm as a basis for our cloud detector. The main problem with using the algorithm as-is was that, persistently, white object-like buildings were masked out in images for all time points, which guaranteed failure of the reconstruction algorithm at these locations. As noted in [46], the most accurate cloud detection algorithms like the state of the art [101] requires temporal sequences. In our augmented model, we simply removed connected regions of clouds and cloud-free regions when these regions had less than 400 pixels, and we also removed pixels that S2Cloudless is marked as cloudy much more frequently than the Sentinel-2 native cloud mask. We generated a cloud mask for every Sentinel-2 tile prior to training and inference with SpaceEye. This leads to a pleasant user experience with very little lag time when running SpaceEye on small space–time regions.

3) Cloud Shadow Detection: We built our own pixelwise cloud-shadow detector using the ExtraTrees classifier [104] provided in Scikit-learn and the Hollstein dataset. The resulting random forest had a high cloud-shadow detection

---

4For the test set we reported results for, only a time step of one day was utilized.
recall (99%) but a low precision. Consequently, we selected only the pixels that were significantly darker (90%) than the average cloud-free luminance for the mean of the same pixel in a window of the 20 temporarily closest Sentinel-2 images. The resulting algorithm fails to detect topological shadows or cloud shadows for periods with a prolonged cloud cover. Our cloud and cloud-shadow detectors are far from state-of-the-art and a potential target for future work.

D. SAR Data Processing

We used radar data from the Sentinel-1 satellite system. Sentinel-1 is a constellation of two satellites, Sentinel-1A and Sentinel-1B, launched into low Earth orbit in April 2014 and 2016, respectively. The satellites are in a 12-day cycle with a 180° orbital phasing difference, giving a six-day repeat frequency. The location revisit rate is three days at the equator and two days in Europe. For our experiments, we used data from the IW swath mode that covers 250 km with dual-polarization VV-VH. The original data need substantial processing to be transformed into an image that can be used in our system. The original pixel resolution is 20 m × 5 m in the azimuth and range directions.

A significant number of steps were needed in order to process data from the Sentinel-1 into images that can be used by SpaceEye. The rest of this section outlines the steps.

1) Data Access: We used the python sentinel3sat module [91] to download recent ground range detected (GRD), dual-polarization Level-1 Sentinel-1 products. For data older than one year, data were acquired through the services of Alaska Satellite Facilities [105].

2) SAR Processing: We used the SNAP toolkit [106] to generate image data from the downloaded data. The processing steps were given as follows.

1) do_apply_orbit_file: This step acquires the satellite orbit file, which is needed later. The precise orbit file is only available 20 days afterward, but processing with a less precise orbit is still possible. We reprocess all images after the more precise orbit file becomes available.

2) do_remove_GRD_border_noise: The Sentinel-1 GRD product has noise artifacts at the image borders, which are quite consistent at both the left- and right-hand sides of the satellite’s cross-track and at the start and end of the data take along the track. The software that we used is described and compared to alternatives in [107].

3) do_thermal_noise_removal: Sentinel-1 image intensity is disturbed by additive thermal noise. Thermal noise removal reduces noise effects in the inter-sub-swath texture [108].

4) do_calibration: Calibration converts pixel values to radiometrically calibrated SAR backscatter.

5) do_terrain_correction: Range Doppler terrain correction rectifies geometric distortions caused by topography, such as foreshortening and shadows. It uses a digital elevation model to correct the location of each pixel. The methods are used to implement the range-Doppler orthorectification method [109]. This step also resamples the resolution to 10 m × 10 m for alignment with corresponding Sentinel-2 products. The default setting for the terrain correction is to remove ocean pixels. We explicitly disabled this to use the same algorithm for both land and sea locations.

6) linear_to_db: This simply brings the signal to the log-domain.

7) do_subset: In this step, we extract the extent of the image that falls inside each intersecting Sentinel-2 tile and apply the corresponding UTM projection.

3) Image Fusion: The Sentinel-1 product contains two bands corresponding to VH and VV polarizations. We stacked the VV and VH bands and compressed the tiff files using the loss-less Z standard (ZSTD) compression algorithm. We also use the TILED = YES option as before. In addition, we merged images along the same orbital sweep and marked pixels outside the image as missing.

4) Image Fusion: The Sentinel-1 product contains two bands corresponding to VH and VV polarizations. We stacked the VV and VH bands and compressed the tiff files using the loss-less Z standard (ZSTD) compression algorithm. We also use the TILED = YES option as before. In addition, we merged images along the same orbital sweep and marked pixels outside the image as missing.

Dataset Characteristics: Sentinel-1 SAR utilizes C-band radar, which partially penetrates canopy, dry sand, and thin layers of snow, while it is reflected off of standing water. As a result, the resulting SAR images give a lot of useful information for agriculture and are ideally suited to detect flooding. The same reflective property also means that changes in watercolor cannot be detected solely from SAR and have to also rely on multispectral data. As a result, our system is not well suited to monitor water quality. Similarly, the snowy ground can be challenging for the system (however, our baselines perform even worse on snowy scenes).

E. Optimization for Damped Interpolation

In this section, we outline the optimization methodology used for damped interpolation in this article. One of our requirements is that the methods should run efficiently to facilitate interactivity. This was achieved mainly by executing the methods on the GPU using pytorch. Recall that the objective function is given by

\[
F(X) = \| M \circ (X - Y) \|_F^2 + \alpha \sum_{t=1}^{T-1} \| X_{t+1} - X_t \|_F^2
\]

where \( Y \in \mathbb{R}^{(C_1 + C_2)T \times HW} \) is the stacked multispectral and radar data, \( M \in \mathbb{R}^{(C_1 + C_2)T \times HW} \) is the data mask (0 for missing data and cloudy pixels, and 1 for clear sky pixels), and \( X \in \mathbb{R}^{(C_1 + C_2)T \times HW} \) is the desired cloud-free reconstruction. The notation \( X_t \) is used to refer to the smaller submatrices of size \((C_1 + C_2) \times HW\) corresponding to the data at time \( t \).

With even the modest choice of dimensions \( C_1 = 10, C_2 = 2, T = 48, H = W = 448 \), \( F \) is a quadratic objective in \( 1.2 \times 10^8 \) variables, which is already too large for a general quadratic solver. We consider a formulation that takes advantage of the problem structure and allows an iterative approach with much less expensive steps. The unconstrained version of this problem (\( M = 1 \)) decouples into separate problems for each pixel and band, and each can be solved separately. We will construct an auxiliary function to take advantage of this.
1) **Damping Coefficient:** The function \( F \) has the interesting property that \( \lim_{\alpha \to 0} \arg\min_X F(X) \) converges to a function that linearly interpolates the temporal sequence of \( Y \) with a constant function as the extrapolation before the first cloud-free pixel and after the last cloud-free pixel. In other words, it is equivalent to a solution that does pixel-based linear interpolation in the time dimension. For \( \alpha > 0 \), it pulls the cloud-free observations closer toward each other, thus the name of the method. We refer to \( \alpha \) as the damping coefficient.

2) **Auxiliary Function:** We use an auxiliary function that gives an upper bound to the objective function \( F \)

\[
Q(X, Z) = \| M \circ (X - Y) \|_F^2 + \|(1 - M) \circ (X - Z)\|_F^2 + \alpha \sum_{t=1}^{T-1} \| X_{t+1} - X_t \|_F^2
\]

\[
= \| X - M \circ Y - (1 - M) \circ (Z) \|_F^2 + \alpha \sum_{t=1}^{T-1} \| X_{t+1} - X_t \|_F^2.
\]

We have \( Q(X, X) = F(X) \) and \( Q(X, Z) \geq F(X) \), and consequently, the global minimum of \( Q \) coincides with that of \( F \). We use an alternate minimization over \( X \) and \( Z \) to iteratively reduce the value of \( Q(X, Z) \).

For fixed \( X \), the minimum over \( Z \) is achieved for \( Z = (1 - M) \circ X \). The minimization over \( X \) for fixed \( Z \) turns out to also be simpler than before. The independent pixel problems can be solved in parallel.

3) **Minimizer:** To describe the solution, define the forward difference matrix \( D \in \mathbb{R}^{T \times T} \) by

\[
D = \begin{pmatrix}
-1 & 1 & 0 & \cdots & 0 & 0 \\
0 & -1 & 1 & \cdots & 0 & 0 \\
0 & 0 & -1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & -1 & 1 \\
0 & 0 & 0 & \cdots & 0 & 0
\end{pmatrix}
\]

(7)

and the companion matrix \( \Delta = I_{C_1+C_2} \otimes D \in \mathbb{R}^{T(C_1+C_2) \times T(C_1+C_2)} \). Fixing \( Z \), the minimum for \( X \) is then given by

\[
X = (I + \alpha \Delta^\top \Delta)^{-1} ((M \circ Y) - (1 - M) \circ Z).
\]

Furthermore, due to the properties of the Kronecker product, we have

\[
(I + \alpha \Delta^\top \Delta)^{-1} = I_{N_r} \otimes (I_{N_r} + \alpha D^\top D)^{-1}.
\]

Since \( I_{N_r} + \alpha D^\top D \) only depends on the dimension \( T \) and \( \alpha \), it can be precomputed so that all calculations are simple matrix multiplications that can easily be handled on the GPU. Putting the resulting two iterations together, we get a single iteration

\[
X \leftarrow (I + \alpha \Delta^\top \Delta)^{-1} ((M \circ Y) - (1 - M) \circ X).
\]

(8)

F. **Optimization for Matrix Completion**

Several approaches for cloud removal have used matrix and tensor completion with nonsmooth optimization [52], [53], [54]. We aimed for simpler, more efficient models and based our work on the methods in [55]. Their approach used the nuclear norm, which, although a convex function, is quite complex and slow to optimize. We simplified it to a low-rank quadratic optimization problem, which was solved with an alternating optimization algorithm implemented on the GPU. Both the damped interpolation and the matrix completion methods are based on the same objective loss function, but the matrix completion method enforces a rank constraint. Wang et al. [55] considered the bands to be independent, but combining bands allows the solution to have a higher rank and benefits from the use of the SAR data.

\[
X \text{ can be decomposed into a low-rank matrix } X = UV^\top
\]

with \( U \in \mathbb{R}^{(C_1+C_2) \times N_r} \) and \( V \in \mathbb{R}^{HW \times N_r} \), where \( N_r \) is the rank. We think of the column vector \( V_r \) as land-type \( r \) and the column vector \( U_r \) as the evolution of the corresponding land-type. Our matrix completion minimizes \( F(UV^\top) \) with respect to \( U \) and \( V \) for a fixed rank \( N_r \). We found the best results with \( N_r = 35 \), which is considerably smaller than the full rank \((C_1 + C_2)T = 576 \) for our test set \((T = 48)\). The optimization algorithm was slower than the damped interpolation and relied on alternating optimization as well.

**Minimizer:** We use the same auxiliary function as in Appendix F and define \( Y_Z = ((M \circ Y) - (1 - M) \circ Z) \). We then optimize \( G(U, V, Z) = Q(UV^\top, Z) \) iteratively with respect to \( Z, V, \text{ and } U \). Since all the equations are quadratic with the other variables fixed, we get closed-form equations

\[
Z = (1 - M) \circ (UV^\top) \quad (9)
\]

\[
U = (I + \alpha \Delta^\top \Delta)^{-1} Y_Z V (V^\top V)^{-1} \quad (10)
\]

\[
V = Y_Z^\top(U(UU^\top + \alpha U^\top \Delta^\top \Delta U)^{-1}) \quad (11)
\]

Note that, with the exception of \((I + \alpha \Delta^\top \Delta)^{-1}\), which can be precomputed, the matrix inverses are of size \( N_r \times N_r \), and we explicitly control the size of \( N_r \). The corresponding loss with the nuclear norm regularizer \( F(X) + \lambda \|X\|_\ast \) can be done in the same manner by using the auxiliary identity \( \|X\|_\ast = \min_{U,V, X = UV^\top} (1/2) (\|U\|_F^2 + \|V\|_F^2) \). However, after the dust settles, the auxiliary formulation for the nuclear norm still requires solving a Sylvester equation that is not available in pytorch for the GPU.

**REFERENCES**

[1] D. J. Muilla, “Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps,” *Biosyst. Eng.*, vol. 114, no. 4, pp. 358–371, Apr. 2013.

[2] T. B. Hank et al., “Spaceborne imaging spectroscopy for sustainable agriculture: Contributions and challenges,” *Surv. Geophys.*, vol. 40, no. 3, pp. 515–551, 2019.

[3] S. Manfreda et al., “On the use of unmanned aerial systems for environmental monitoring,” *Remote Sens.*, vol. 10, no. 4, p. 641, 2018.

[4] C. J. Tucker and J. R. G. Townshend, “Strategies for monitoring tropical deforestation using satellite data,” *Int. J. Remote Sens.*, vol. 21, nos. 6–7, pp. 1461–1471, Jan. 2000.

[5] D. M. Tralli, R. G. Blom, V. Zlotnicki, A. Donnellan, and D. L. Evans, “Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards,” *ISPRS J. Photogramm. Remote Sens.*, vol. 59, no. 4, pp. 185–198, 2005.

[6] Y. Cunjun, W. Yiming, W. Siyuan, Z. Zengxiang, and H. Shifeng, “Extracting the flood extent from satellite SAR image with the support of topographic data,” in *Proc. Int. Conf. Info-Tech Info-Net.*, vol. 1, Oct./Nov. 2001, pp. 87–92.
D.-C. Tseng, H.-T. Tseng, and C.-L. Chien, “Automatic cloud removal

P. Singh and N. Komodakis, “Cloud-GAN: Cloud removal for Sentinel-

J. Dong, R. Yin, X. Sun, L. Li, Y. Yang, and X. Qin, “Inpainting

M. Xu, X. Jia, M. Pickering, and A. J. Plaza, “Cloud removal using a cyclic consistent generative adversarial networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops, Jul. 2017, pp. 48–56.

J. Dong, R. Yin, X. Sun, L. Li, Y. Yang, and X. Qin, “Inpainting of remote sensing SST images with deep convolutional generative adversarial network,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 2, pp. 173–177, Feb. 2019.

P. Singh and N. Komodakis, “Cloud-GAN: Cloud removal for Sentinel-2 imagery using a cyclic consistent generative adversarial networks,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Jul. 2018, pp. 1772–1775.

K.-Y. Lee and J.-Y. Sim, “Cloud removal of satellite images using convolutional neural network with reliable cloudy image synthesis model,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2019, pp. 3581–3585.

J. Zheng, X.-Y. Liu, and X. Wang, “Single image cloud removal using U-Net and generative adversarial networks,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 8, pp. 6371–6385, Aug. 2021.

D.-C. Tseng, H.-T. Tseng, and C.-L. Chien, “Automatic cloud removal from multi-temporal SPOT images,” Appl. Math. Comput., vol. 205, no. 2, pp. 584–600, Nov. 2008.

M. Xu, X. Jia, M. Pickering, and A. J. Plaza, “Cloud removal based on sparse representation via multitemporal dictionary learning,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 5, pp. 2998–3006, May 2016.

V. Sarukkai, A. Jain, B. Uzkent, and S. Ermon, “Cloud removal from satellite images using spatiotemporal generator networks,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., Mar. 2020, pp. 1796–1805.

O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, “Image inpainting: A review,” Neuro Process. Lett., vol. 51, pp. 2007–2028, Dec. 2019.

D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, “Deep video inpainting,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5792–5801.

J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5505–5514.

J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, “Free-form image inpainting with gated convolution,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4471–4480.

G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 85–100.

S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” ACM Trans. Graph., vol. 36, no. 4, pp. 1–14, 2017.

D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2536–2544.

R. Xu, X. Li, B. Zhou, and C. C. Loy, “Deep flow-guided video inpainting,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3723–3732.

F. Nunziata, X. Li, M. Migliaccio, A. Montuori, and W. Pichler, “Metallic objects and oil spill detection with multi-polarization SAR,” in Proc. IEEE Int. Geosci. Remote Sens. Symp., Jul. 2010, pp. 2765–2768.

P. Mastro, G. Masiello, C. Serio, and A. Pepe, “Change detection techniques with synthetic aperture radar images: Experiments with random forests and Sentinel-1 observations,” Remote Sens., vol. 14, no. 14, p. 3323, Jul. 2022.

S. Schlaffer, P. Matgen, M. Hollaus, and W. Wagner, “Flood detection from multi-temporal SAR data using harmonic analysis and change detection,” Int. J. Appl. Earth Observ. Geoinf., vol. 38, pp. 15–24, Jun. 2015.

N. Kussul, A. Shelestov, and S. Skakun, “Flood monitoring from SAR data,” in Use of Satellite and In-Situ Data to Improve Sustainability, F. Kogan, A. Powell, and O. Fedorov, Eds. Dordrecht, The Netherlands: Springer, 2011, pp. 19–29.

R. Michel, J. P. Avouac, and J. Taboury, “Measuring ground displacements from SAR amplitude images: Application to the Landers earthquake,” Geophys. Res. Lett., vol. 26, no. 7, pp. 875–878, Apr. 1999.

W. Knöpfle, G. Strunz, and A. Roth, “Mosaicking of digital elevation models derived by SAR interferometry,” Int. Arch. Photogramm. Remote Sens., vol. 32, pp. 306–313, Mar. 1998.

L. Wang et al., “SAR-to-optical image translation using supervised cycle-consistent adversarial networks,” IEEE Access, vol. 7, pp. 129136–129149, Dec. 2019.

A. Meraner, P. Ebel, X. X. Zhu, and M. Schmitt, “Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion,” ISPRS J. Photogramm. Remote Sens., vol. 166, pp. 333–346, Aug. 2020.

Copernicus Open Access Hub. Accessed: May 20, 2021. [Online]. Available: https://scihub.copernicus.eu/

Sentinel Missions. Accessed: May 20, 2021. [Online]. Available: https://sentinel.esa.int/web/sentinel/missions

S. Foga et al., “Cloud detection algorithm comparison and validation for operational Landsat data products,” Remote Sens. Environ., vol. 194, pp. 379–390, Jun. 2017.

A. H. Sanchez et al., “Comparison of Cloud cover detection algorithms on Sentinel-2 images of the Amazon tropical forest,” Remote Sens., vol. 12, no. 8, p. 1284, 2020.

T. Bai, D. Li, K. Sun, Y. Chen, and W. Li, “Cloud detection for high-resolution satellite imagery using machine learning and multi-feature fusion,” Remote Sens., vol. 8, no. 9, p. 715, 2016.

J. H. Jeppesen, R. H. Jacobsen, F. Incocglo, and T. S. Toltegaard, “A cloud detection algorithm for satellite imagery based on deep learning,” Remote Sens. Environ., vol. 229, pp. 247–259, May 2019.

A. Zebker and C. E. Werner, “Object-based cloud and cloud shadow detection in Landsat imagery,” Remote Sens. Environ., vol. 118, pp. 83–94, Mar. 2012.

L. Baetens, C. Desjardins, and O. Hagolle, “Validation of Copernicus Sentinel-2 cloud masks obtained from MAJA, Sen2Cor, and FMask processors using reference cloud masks generated with a supervised active learning procedure,” Remote Sens., vol. 11, no. 4, p. 433, Feb. 2019.

F. Meng, X. Yang, C. Zhou, and Z. Li, “A sparse dictionary learning-based adaptive patch inpainting method for thick clouds removal from high-resolution remote sensing imagery,” Sensors, vol. 17, no. 9, p. 2130, Sep. 2017.

Y. Li, W. Li, and C. Shen, “Removal of optically thick clouds from high-resolution satellite imagery using dictionary group learning and interdictionary nonlocal joint sparse coding,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 10, no. 5, pp. 1870–1882, May 2017.

E. H. Helmer and B. Ruefenacht, “Cloud-free satellite image mosaics with regression trees and histogram matching,” Photogramm. Eng. Remote Sens., vol. 71, no. 9, pp. 1079–1089, 2005.

C. Lai, P. H. Tsai, and Y. C. Lin, “Cloud removal from multitemporal satellite images using information cloning,” IEEE Trans. Geosci. Remote Sens., vol. 51, no. 1, pp. 232–241, Jan. 2013.

F. Vuolo, W.-T. Ng, and C. Atzberger, “Smoothing and gap-filling of high resolution multi-spectral time series: Example of Landsat data,” Int. J. Appl. Earth Observ. Geoinf., vol. 57, pp. 202–213, May 2017.

T.-Y. Ji, N. Yokoya, X. X. Zhu, and T.-Z. Huang, “Nonlocal tensor completion for multitemporal remotely sensed images’ inpainting,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 6, pp. 3047–3061, Jun. 2018.
A. Aravkin, S. Becker, V. Cevher, and P. Olsen, “A variational approach to stable principal component pursuit,” 2014, arXiv:1406.1089.

Y. Ma et al., “Fusion of multi-angular aerial images based on epipolar geometry and matrix completion,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2017, pp. 1197–1201.

J. Wang, P. A. Olsen, A. R. Conn, and A. C. Lozano, “Removing clouds and recovering ground observations in satellite image sequences via temporally consistent robust matrix completion,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2754–2763.

K. Hazaymeh and Q. K. Hassan, “Spatiotemporal image-fusion model for enhancing the temporal resolution of Landsat-8 surface reflectance images using MODIS data,” J. Appl. Remote Sens., vol. 9, no. 1, Jun. 2015, Art. no. 096065.

Y. Luo, K. Gao, and J. Peng, “STAIR: A generic and fully-adaptive method to fuse multiple sources of optical satellite data to generate a high-resolution, daily and cloud-/gap-free surface reflectance product,” Remote Sens. Environ., vol. 214, pp. 87–99, Sep. 2018.

R. Houborg and M. F. McCabe, “A cubestat enabled spatio-temporal enhancement method (CESTEM) utilizing planet, Landsat and MODIS data,” Remote Sens. Environ., vol. 209, pp. 211–226, May 2018.

C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, “Patch-Match: A randomized correspondence algorithm for structural image editing,” ACM Trans. Graph., vol. 28, no. 3, p. 24, Jul. 2009.

M. Strobel, J. Diebold, and D. Cremers, “Flow and color inpainting for video completion,” in Pattern Recognition, X. Jiang, J. Hornegger, and R. Koch, Eds. Cham, Switzerland: Springer, 2014, pp. 293–304.

J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, “Temporally coherent completion of dynamic video,” ACM Trans. Graph., vol. 35, no. 6, pp. 1–11, Nov. 2016.

H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, “The sound of pixels,” in Proc. Eur. Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.

R. Owens and A. A. Efros, “Audio-visual scene analysis with self-supervised multisensory features,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 631–648.

R. Gao, R. Feris, and K. Grauman, “Learning to separate object sounds by watching unlabeled video,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 35–53.

H. Zhao, C. Gan, W.-C. Ma, and A. Torralba, “The sound of motions,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 1737–1744.

M. Zhao et al., “RF-based 3D skeletons,” in Proc. Conf. ACM Special Interest Group Data Commun., 2018, pp. 267–281.

F. Zheng, “Learning the signatures of the human grasp using a scalable tactile glove,” J. Sensonic., vol. 40, no. 7, Jul. 2019, Art. no. 070202.

A. Salvador et al., “Learning cross-modal embeddings for cooking recipes and food images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 3020–3028.

M. Zhao et al., “Through-wall human pose estimation using radio signals,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7356–7365.

T. Li, L. Fan, M. Zhao, Y. Liu, and D. Katabi, “Making the invisible visible: Action recognition through walls and occlusions,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 872–881.

M. Zhao et al., “Through-wall human mesh recovery using radio signals,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 10113–10122.

M. Soumekh, Synthetic Aperture Radar Signal Processing, vol. 7, New York, NY: Wiley, 1990.

S. Issef and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. 32nd Int. Conf. Mach. Learn., vol. 37, 2015, pp. 448–456.

Z. Houborg and M. F. McCabe, “A cubesat enabled spatio-temporal enhancement method (CESTEM) utilizing planet, Landsat and MODIS data,” Remote Sens. Environ., vol. 209, pp. 211–226, May 2018.

J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, “Temporally coherent completion of dynamic video,” ACM Trans. Graph., vol. 35, no. 6, pp. 1–11, Nov. 2016.

H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, “The sound of pixels,” in Proc. Eur. Conf. Comput. Vis. Pattern Recognit., Jun. 2008, pp. 1–8.

R. Owens and A. A. Efros, “Audio-visual scene analysis with self-supervised multisensory features,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 631–648.

R. Gao, R. Feris, and K. Grauman, “Learning to separate object sounds by watching unlabeled video,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 35–53.

H. Zhao, C. Gan, W.-C. Ma, and A. Torralba, “The sound of motions,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 1737–1744.

M. Zhao et al., “RF-based 3D skeletons,” in Proc. Conf. ACM Special Interest Group Data Commun., 2018, pp. 267–281.

F. Zheng, “Learning the signatures of the human grasp using a scalable tactile glove,” J. Sensonic., vol. 40, no. 7, Jul. 2019, Art. no. 070202.

A. Salvador et al., “Learning cross-modal embeddings for cooking recipes and food images,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 3020–3028.

M. Zhao et al., “Through-wall human pose estimation using radio signals,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7356–7365.

T. Li, L. Fan, M. Zhao, Y. Liu, and D. Katabi, “Making the invisible visible: Action recognition through walls and occlusions,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 872–881.

M. Zhao et al., “Through-wall human mesh recovery using radio signals,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., Oct. 2019, pp. 10113–10122.

M. Soumekh, Synthetic Aperture Radar Signal Processing, vol. 7, New York, NY: Wiley, 1990.

S. Issef and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in Proc. 32nd Int. Conf. Mach. Learn., vol. 37, 2015, pp. 448–456.
[106] *Science Toolbox Exploration Platform—SNAP Download.* Accessed: Aug. 23, 2020. [Online]. Available: http://step.esa.int/main/download/snap-download/

[107] I. Ali, S. Cao, V. Naeimi, C. Paulik, and W. Wagner, “Methods to remove the border noise from Sentinel-1 synthetic aperture radar data: Implications and importance for time-series analysis,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 777–786, Mar. 2018.

[108] F. Filipponi, “Sentinel-1 GRD preprocessing workflow,” *Proceedings*, vol. 18, no. 1, p. 11, 2019.

[109] D. Small and A. Schubert, “Guide to ASAR geocoding,” ESA-ESRIN, Frascati, Italy, Tech. Note RSL-ASAR-GC-AD, 2008, vol. 1, p. 36.

*Mingmin Zhao* (Member, IEEE) received the Ph.D. degree from the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2021. He is currently an Assistant Professor with the University of Pennsylvania, Philadelphia, PA, USA. His research interests include wireless sensing and machine learning.

Dr. Zhao was a recipient of the ACM SIGMOBILE Doctoral Dissertation Award Runner Up, the ACM SIGMOBILE Research Highlights, the ACM Research Highlights, the Baidu Fellowship, and the Yunfan award for Rising Stars in AI.

*Peder Olsen* (Senior Member, IEEE) received the Ph.D. degree in mathematics from the University of Michigan, Ann Arbor, MI, USA, in 1996. He is a Principal Research Scientist in research for the industry with Microsoft, Redmond, WA, USA. Prior to joining Microsoft Research, he was a Principal Research Staff Member at IBM TJ Watson Research Center (Yorktown Heights, NY), working in the speech recognition group. His research focuses on AI, machine learning, and computer vision applications of remote sensing.

*Ranveer Chandra* (Fellow, IEEE) is currently the CTO of Agri-Food and the Managing Director of Research for Industry of Microsoft, Redmond, WA, USA. His research focuses on networking and systems.

Dr. Chandra received multiple best paper awards and recognitions, such as the MIT Technology Review Top Innovator Under 35 (TR-35) and the America’s 50 Most Disruptive Innovators by the *Newsweek* magazine.