AGGREGATING CLOUD-FREE SENTINEL-2 IMAGES WITH GOOGLE EARTH ENGINE

Michael Schmitt[1], Lloyd H. Hughes[1], Chunping Qiu[1], Xiao Xiang Zhu[1,2]

[1] Signal Processing in Earth Observation, Technical University of Munich, Munich, Germany
- (m.schmitt, lloyd.hughes, chunping.qiu)@tum.de
[2] Remote Sensing Technology Institute, German Aerospace Center (DLR), Oberpfaffenhofen, Wessling - xiaoxiang.zhu@dlr.de

KEY WORDS: Sentinel-2, Google Earth Engine, Cloud Coverage, Big Data

ABSTRACT:

Cloud coverage is one of the biggest concerns in spaceborne optical remote sensing, because it hampers a continuous monitoring of the Earth's surface. Based on Google Earth Engine, a web- and cloud-based platform for the analysis and visualization of large-scale geospatial data, we present a fully automatic workflow to aggregate cloud-free Sentinel-2 images for user-defined areas of interest and time periods, which can be significantly shorter than the one-year time frames that are commonly used in other multi-temporal image aggregation approaches. We demonstrate the feasibility of our workflow for several cities spread around the globe and affected by different amounts of average cloud cover. The experimental results confirm that our results are better than the results achieved by standard approaches for cloud-free image aggregation.

1. INTRODUCTION

As determined by the MODIS mission, on average, about 67% of the Earth's surface are covered by clouds (King et al., 2013) (cf. Figure 1), posing a well-known drawback for any remote sensing endeavours aiming at a monitoring of the Earth's surface and relying on sensors operating in the optical domain. In order to avoid the information gaps caused by clouds, Earth observation traditionally either resorts to sensors operating in the microwave domain or to algorithmic cloud removal strategies. These are usually based on interpolation methods (Cihlar and Howarth, 1994, Zhu et al., 2012), machine-learning-based void filling approaches (Cheng et al., 2014, Chang et al., 2015, Huang et al., 2015, Xu et al., 2016), exploiting multi-sensor data fusion (Huang et al., 2015) or multi-temporal image sets (Lin et al., 2013, Cheng et al., 2014, Xu et al., 2016, Candra et al., 2017). However, all these approaches have different drawbacks: In the case of data fusion, the joint availability of complementary data sources (e.g. a synthetic aperture radar (SAR) image to fill in missing information in a cloud-affected optical image) needs to be ensured, while void filling approaches make up data based on constraints learned from the internal data structure. Finally, most methods exploiting multi-temporal imagery usually rely on rather long time series, e.g. about 1 year for a global cloud-free Sentinel-2 mosaic (May 2016 to April 2017) (Sentinel-2 cloudless, 2017) or a cloud-free Sentinel-2 mosaic of the southern extent of the African continent (January 2016 to December 2016) (Ramoino et al., 2017). In these cases, temporal stability of the land cover cannot be ensured, which renders multi-temporal cloud-free mosaics an inadequate resource for fine-grained monitoring or change detection approaches.

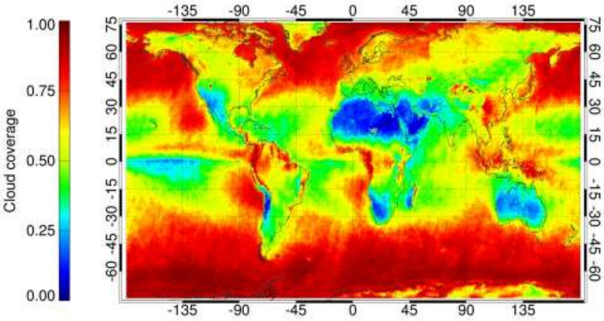


Figure 1. Mean annual cloud coveras observed by the Envisat mission in the years 2007-2009. Image: ESA/Cloud-CCI.

With this paper, we propose a cloud-based engineering approach that allows to aggregate – and export – (mostly) cloud-free Sentinel-2 multi-spectral images for rather concise time windows using Google Earth Engine (GEE). The method relies both on pixel-wise cloud detection as well as the combination of multi-temporal information of comparably short time periods – we have chosen the meteorological seasons as time frames in order to be able to produce multi-seasonal images for arbitrary regions of interest. The strengths of the approach are manifold:

- It does not infer pixels based on statistical or machine learning models but makes use of posteriori information which was actually measured by Sentinel-2.
- While being able to generate *mostly* cloud-free images even for severely cloud-affected regions of interest (ROIs), the method always strives to create images that are as clean and artifact-free as possible.
- Using GEE's cloud computation infrastructure, it can efficiently produce cloud-free images for large numbers of ROIs and time frames in a parallel manner.

In order to document the methodology and prove its usefulness, the remainder of this paper is structured as follows: Section 2 describes our workflow implemented in form of individual processing modules in Google Earth Engine. Section 3 illustrates several example cases for cloud-free image production for areas with different amount of cloud coverage. Finally, Section 4 discusses the achieved results as well as the advantages and drawbacks of the

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W7-145-2019

proposed method, before Section 5 summarizes and concludes our work.

2. GOOGLE EARTH ENGINE-BASED WORKFLOW FOR CLOUD-FREE SENTINEL-2 IMAGE GENERATION

Google Earth Engine (GEE) is a web- and cloud-based platform for large-scale scientific analysis and visualization of geospatial data. It provides an extensive catalogue of remote sensing imagery and other geodata, as well as an application programming interface (API) with both JavaScript and Python front-ends allowing for the analysis of the data available in the catalogue on Google's servers (Gorelick et al., 2017). The overall workflow, which we implemented using the GEE Python API[1], is depicted in Figure 2. We also made a Javascript version available via the GEE platform[2]. In essence, it consists of three main modules: (1) The *Query Module* for loading images from the catalogue, (2) the *Quality Score Module* for the calculation of a quality score for each image, and (3) the *Image Merging Module* for mosaicking of the selected images based on the meta-information generated in the preceding modules. All these modules are described in detail the following subsections.

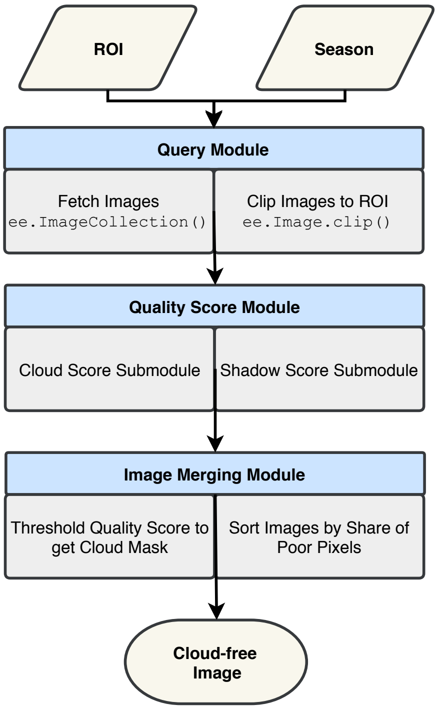


Figure 2. Overall workflow of the GEE-based procedure for cloud-free Sentinel-2 image generation presented in this paper. During the process several sub-modules are called.

2.1 Query Module

Our data preparation workflow starts by reading in the list of regions of interest (ROIs) from a Google fusion table into a GEE feature collection using the command `ee.FeatureCollection()`. In addition, the desired time frame for which a cloud-free image is to be created, needs to be defined. While extended time frames (say about a year, for example) will allow us to produce cloud-free images by mosaicking multi-temporal data, the resulting image might contain observations from different seasons and thus contain inhomogeneous radiometric information. On the other hand, very narrow time frames (e.g. a single day or week) will sometimes not contain any cloud-free pixels for the ROI. Thus, a reasonable trade-off has to be found.

The image collection is then filtered for images acquired in the defined time frame by `ee.ImageCollection.filterDate()` and the images are clipped to the ROI to reduce storage requirements and processing time using `ee.Image.clip()`. The resulting image collection is then put into the actual workflow comprised of the remaining modules.

2.2 Quality Score Module

The second module aims at the calculation of a quality score (QS) for each pixel in each image. This quality score is later used to determine the image pixel selection in order to create the cloud free image.

The QS is calculated from the cloud score (CS) and shadow score (SS) layers. The negative of the maximum value of the CS and SS is selected as the quality score, thus ensuring that both shadow and cloud are treated equally when selecting the final image pixels. The negative is used as the score should measure the 'goodness' of a pixel and thus needs to be inverted.

The submodules for CS and SS calculation are described in the following.

2.2.1 Cloud Score Submodule The flowchart for the cloud score computation is shown in Figure 3. We have basically adapted the `ee.Algorithms.Landsat.simpleCloudScore()` routine of the GEE API to the Sentinel-2 case (Candra et al., 2017). This adaption is achieved by selecting the appropriate bands from Sentinel 2 to align with the original Landsat bands, and adjusting the classification thresholds are adjusted to account for theses differences. The principle of the algorithm is to recognize that clouds are bright, moist and not the same as snow. In order to implement this, each image starts with an initial cloud map where the cloud score values of each pixel are set to 1, which indicates full cloud coverage. Then, for each pixel the cloud score value is set to the minimum of the previous cloud score and the following values in a sequential manner:

- Blue (band 2), rescaled range $[0.1; 0.5]$
- Aerosol (band 1), rescaled range $[0.1; 0.3]$
- Cirrus+Aerosol (band 10), rescaled range $[0.5; 0.7]$
- Red+Green+Blue (bands 4,3,2) rescaled range $[0.2; 0.8]$
- $NDMI = \frac{NIR(band8)-SWIR(band11)}{NIR(band8)+SWIR(band11)}$, rescaled range $[-0.1; 0.1]$
- $NDSI = \frac{Green(band3)-SWIR(band11)}{Green(band3)+SWIR(band11)}$, rescaled range $[0.8; 0.6]$

NDMI refers to the normalized difference moisture index (Gao, 1996), NDSI to the normalized difference snow index (Hall and Riggs, 2011), and the rescale operation remaps the pixels in the specified range to $[0.0, 1.0]$ in a linear manner, effectively stretching the boundaries of the image to allow more fine grained selectivity. The cloud scoring and rescaling function is defined in

[1] https://github.com/google/earthengine-api
[2] http://bit.ly/Sen2CloudFree

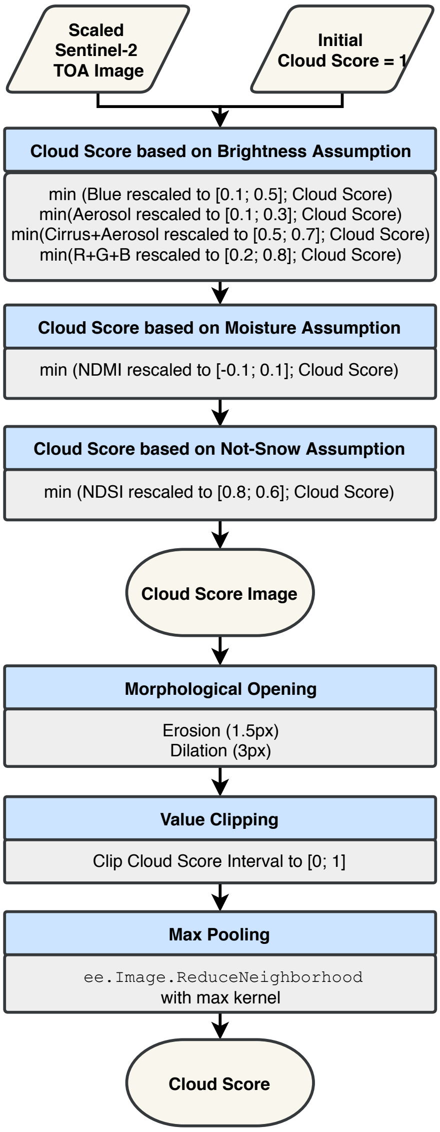

Scaled Sentinel-2 TOA Image
Initial Cloud Score = 1
Cloud Score based on Brightness Assumption
min (Blue rescaled to [0.1; 0.5]; Cloud Score)
min(Aerosol rescaled to [0.1; 0.3]; Cloud Score)
min(Cirrus+Aerosol rescaled to [0.5; 0.7]; Cloud Score)
min(R+G+B rescaled to [0.2; 0.8]; Cloud Score)
Cloud Score based on Moisture Assumption
min (NDMI rescaled to [-0.1; 0.1]; Cloud Score)
Cloud Score based on Not-Snow Assumption
min (NDSI rescaled to [0.8; 0.6]; Cloud Score)
Cloud Score Image
Morphological Opening
Erosion (1.5px)
Dilation (3px)
Value Clipping
Clip Cloud Score Interval to [0; 1]
Max Pooling
ee.Image.ReduceNeighborhood
with max kernel
Cloud Score

Figure 3. Flowchart of the *Cloud Score* module, which produces both a cloud score image containing pixel-wise cloud scores, as well as a scalar cloud score characterizing the whole image.

Equ. 1, where $\mathbf{S}$ represents the current cloud score image and the rescaling range is defined as $[a, b]$.

$$\mathbf{S} = \min\left(\max\left(\frac{\mathbf{S}-a}{b-a}, 0\right), \mathbf{S}\right) \tag{1}$$

The result is a *Cloud Score Image*, which contains a cloud score per pixel. The higher the score, the more likely it is that a pixel is containing only cloud information.

Then, morphological opening and closing is applied to the *Cloud Score Image*. The opening operation is applied first in order to remove single pixels with a high cloud score, which are in the neighborhood of pixels with low cloud scores. These single pixel clouds are often correlated with building rooftops which have a high reflectance. The closing operation is then used to fill any holes which occur in areas with a high cloud score and to ensure edge regions of clouds are correctly scored.

Subsequently, all cloud score pixel values are clipped to the interval $[0; 1]$ before a maximum kernel filter is applied (implemented as ee.Image.ReduceNeighborhood in GEE) to create the final, smoothed, pixel-wise *Cloud Score Image*. An example of this image can be seen in Figure 7b.

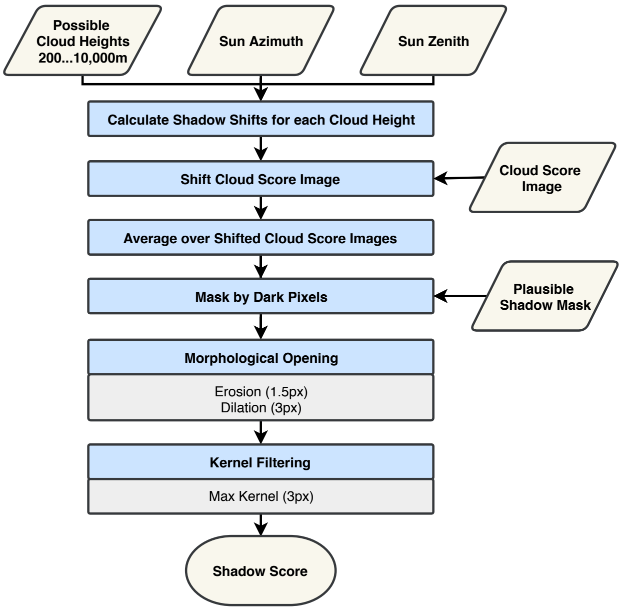


Figure 4. Flowchart of the *Shadow Score* module, which produces a scalar cloud score characterizing the whole image regarding the amount of pixels affected by cloud shadows.

2.2.2 Shadow Score Submodule

The flowchart for the computation of the shadow score is shown in Figure 4. It uses image metadata (sun azimuth and zenith), the previously computed *Cloud Score Image*, and a range of possible shadow heights as input in order to calculate the expected positions of cloud shadow on the ground. To avoid confusion between pixels appearing dark because of dark materials with shadow pixels, only image regions contained in a corresponding plausible shadow mask are considered in the shadow pixel detection. The workflow for the generation of this *Plausible Shadow Mask* is illustrated in Figure 5. In order to calculate the *Plausible Shadow Mask*, the sum of bands 1 (aerosoles), 11, and 12 (both short-wave infrared) are summed and thresholded to select pixels with low reflectance. These pixels, which also have a low normalized difference vegetation index $NDVI = \frac{NIR-R}{NIR+R}$, are then discarded in order to remove dark pixels which are likely due to water bodies. Finally the set of plausible shadow pixels is masked to exclude any pixels which coincide with cloud pixels, determined by thresholding the *Cloud Score Image*, in order to create the *Plausible Shadow Mask*.

Using the *Plausible Shadow Mask* as additional input, the possible shadows cast by the clouds represented in the cloud score image are projected to the ground level, averaged, and then intersected with the plausible shadow locations. Then, morphological erosion and dilation is applied to the resulting intermediate map and kernel filtering is applied in order to retrieve the shadow score for the respective image.

2.3 Image Merging Module

After the *Quality Score* has been calculated, by selecting the maximum value between the *Cloud Score* and *Shadow Score* for each pixel, we threshold the score in order to create a binary classification of bad pixels (i.e. pixels affected by shadow or cloud) and good pixels. This classification layer is used to determine the

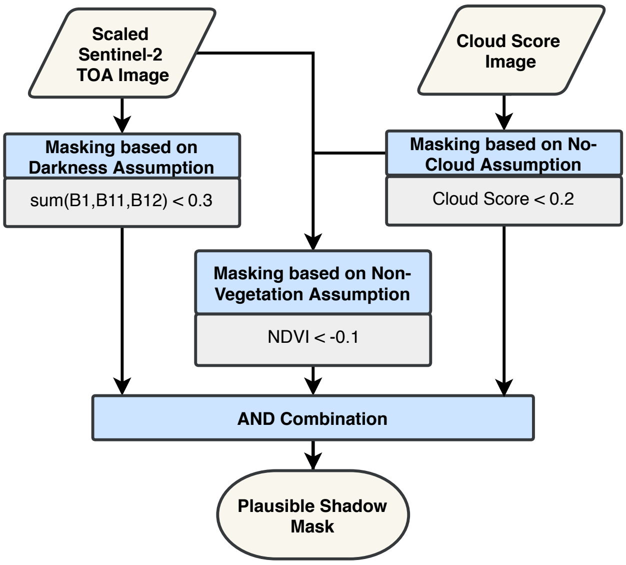

Scaled Sentinel-2 TOA Image
Cloud Score Image
Masking based on Darkness Assumption
sum(B1,B11,B12) < 0.3
Masking based on No-Cloud Assumption
Cloud Score < 0.2
Masking based on Non-Vegetation Assumption
NDVI < -0.1
AND Combination
Plausible Shadow Mask

Figure 5. Flowchart of the *Plausible Shadow Mask* submodule, which is needed to provide input to the *Shadow Score* submodule.

percentage of bad pixels in each image. Using this bad pixel percentage the image collection is sorted in descending order such that the image with the worst score is on top. In Figure 7c we can see an example of the result of thresholding the *Quality Score* layer in the form of a bad pixel mask.

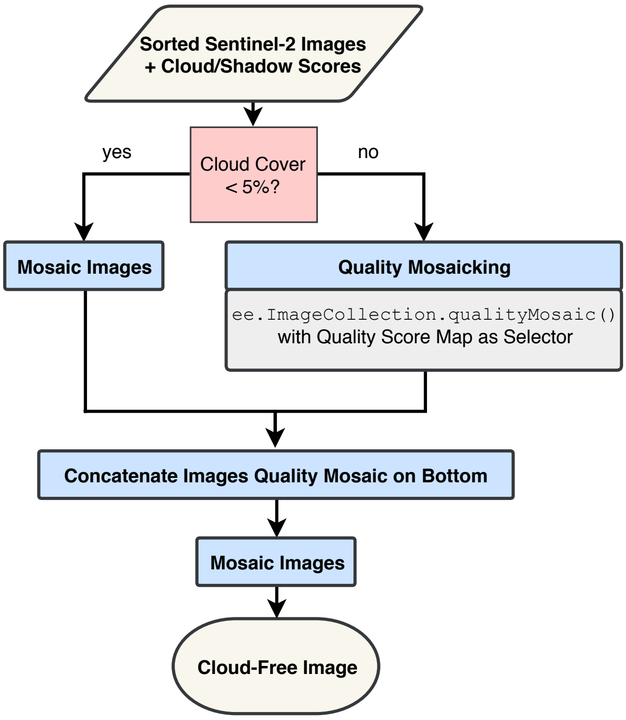

Sorted Sentinel-2 Images + Cloud/Shadow Scores
yes
Cloud Cover < 5%?
no
Mosaic Images
Quality Mosaicking
ee.ImageCollection.qualityMosaic()
with Quality Score Map as Selector
Concatenate Images Quality Mosaic on Bottom
Mosaic Images
Cloud-Free Image

Figure 6. Flowchart of the *Image Merging* module, which finally creates the cloud-free Sentinel-2 image from the pre-processed image collection.

Finally, image merging takes place in order to produce the final cloud free image. The main concept behind the *Image Merging Module* is depicted in Figure 6: If images with less than 5% of bad pixels are available in the collection, we simply use these images to produce the final image using the `ee.ImageCollection.mosaic()` function. If, however, no image with less than 5% of bad pixels is found, we apply the `ee.ImageCollection .qualityMosaic()` function with the *Quality Score Image* as the quality indicator.

While `ee.ImageCollection.mosaic()` just composes all images in an image collection following a *last-on-top* fashion, `ee .ImageCollection.qualityMosaic()` uses a quality indicator, in our case the *Quality Score*, to select which image to use for as the pixel source for each pixel in the final mosaic. Should the area of interest, for which a cloud-free image is to be produced contain more than one Sentinel-2 granule, it can happen that not the entire region of interest (ROI) is covered by cloud free granules. In this case we concatenate the partial cloud free image with the quality mosaic image and mosaic these two images together in order to fill in any gaps and ensure the entire ROI is covered. This image can then be exported to Google Cloud Storage using the API function `ee.batch.Export.image.toCloudStorage()`, the final result of which is depicted in Figure 7d.

In the context of the *Image Merging Module*, it has to be noted that a precise co-registration of the utilized multi-temporal images is of crucial importance. As confirmed by the Sentinel-2 L1C Data Quality Report (European Space Agency, 2019), 98% of all Sentinel-2 products show a multi-temporal registration accuracy better than 1.5 10m-pixels, which is likely to improve in the future.

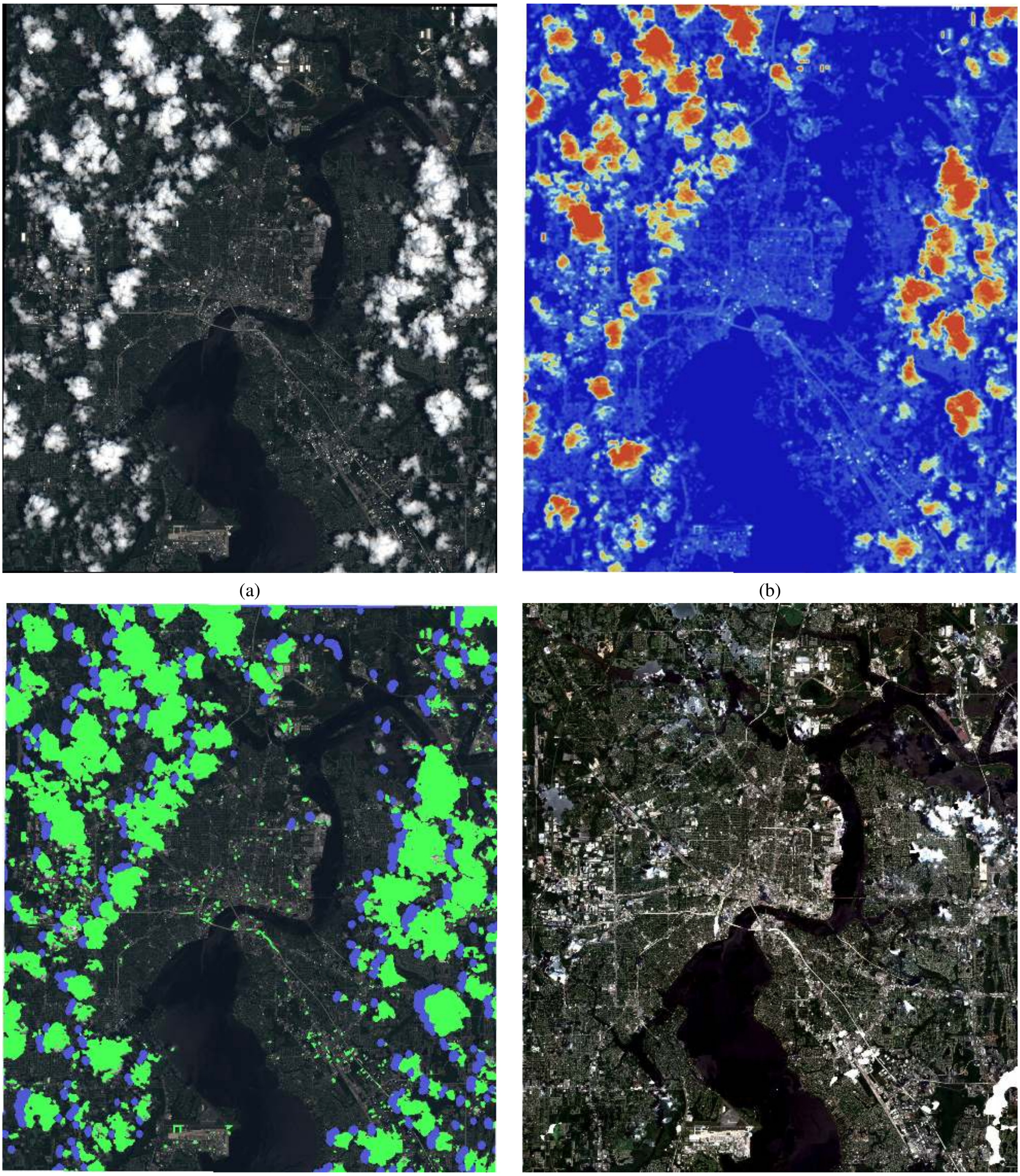
3. VALIDATION OF THE METHOD

To evaluate and validate our image aggregation method, we aim at generating cloud-free images for time spans which reflect the meteorological seasons of the northern hemisphere. The corresponding times are summarized in Table 1. We consider *seasons* a reasonable trade-off between a time frame that is significantly shorter than one year used in standard multi-temporal image aggregation approaches (Sentinel-2 cloudless, 2017, Ramoino et al., 2017), but still long enough to have the chance to gather at least some cloud-free input information. Furthermore, for areas that are affected by seasonal land cover changes, we assume intra-seasonal changes to be less significant than inter-seasonal changes. Last, but not least, there have been first hints in the literature that a fusion of multi-seasonal information can already provide a useful information gain for land cover classification approaches (Qiu et al., 2019).

Table 1. Meteorological seasons as defined for the northern hemisphere.

Season	*Time period*
Spring	01 March till 31 May
Summer	01 June till 31 August
Autumn	01 September till 30 November
Winter	01 December till 28/29 February

For our experiments, we have chosen examples from four different categories: (1) areas that are rarely covered by clouds, (2) areas that are moderately covered by clouds, (3) areas that are frequently covered by clouds, and (4) areas that are almost always covered by clouds. The results achieved for the examples from those categories are described in the following sections. For all examples, we compare the results achieved by our *full framework* and by selecting only the *least cloudy image* based on the bad pixel percentage to two standard procedures: calculation of a *median image*, where each pixel represents the median value of the



(a)

(b)

(c)

(d)

Figure 7. The cloud-free image generation process. (a) The original image from the collection with the least cloud cover for Jacksonville, Florida in winter, (b) computed cloud score for the image, with a color scale from blue (low cloud probability) to red (high cloud probability), (c) cloud and shadow mask computed by thresholding the *Quality Score* with green representing the cloud contribution and blue the shadow, and (d) the final cloud-free image produced for the scene.

multi-temporal samples per band, and a *greenest pixel mosaic*, where the final pixel values are selected based on the temporal sample with the highest NDVI value.

3.1 Rarely cloud-affected areas

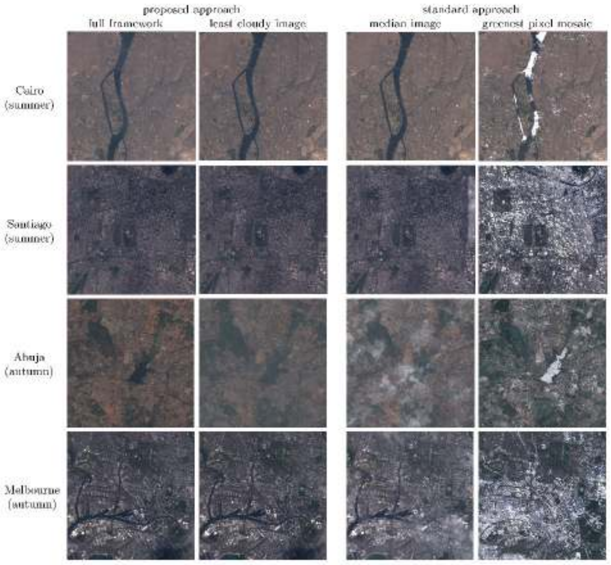


Figure 8. Cloud-free image aggregation examples for areas rarely affected by clouds.

Figure 8 compares the results achieved by our full approach and the least cloudy image, based on the lowest bad pixel percentage (two leftmost columns) to the two standard approaches (two rightmost columns) for individual, relatively cloud-free seasons of the cities of Cairo, Santiago, Abuja, and Melbourne. It can be seen that our full approach provides the overall best images with the least cloudy image following closely behind. In contrast, the simple median approach leaves some clouds in the Santiago, Abuja and Melbourne cases, while the greenest pixel mosaic introduces some artifacts mainly over water areas.

3.2 Moderately cloud-affected areas

Figure 9 compares the results achieved by our full approach and the least cloudy image based on the lowest bad pixel percentage (two leftmost columns) to the two standard approaches (two rightmost columns) for individual, moderately cloud-affected seasons of the cities of Munich, Moscow, Nairobi, and Washington. It becomes apparent that our approach always choses the least cloudy image in order to avoid multi-temporal data aggregation and the introduction of corresponding artifacts. While the greenest pixel mosaic provides a useful solution for Nairobi, its results for Munich, Moscow and Washington are not acceptable due to artifacts for water and large road surfaces. The same holds for all results provided by classical median-based image aggregation, which is not capable of removing clouds anymore.

3.3 Frequently cloud-affected areas

Figure 10 compares the results achieved by our full approach and the least cloudy image based on the lowest bad pixel percentage (two leftmost columns) to the two standard approaches (two rightmost columns) for individual, often cloud-affected seasons of the cities of Bogota, Jakarta, Singapore, and Mumbai. While our full approach introduces some aggregation artifacts to all cloud-free result images, and even leaves some small remaining clouds for Jakarta, its results are significantly better than all other approaches. Even the least cloudy image is not a viable choice for such cases where the area of interest is almost always covered by at least some clouds. Still, the least-cloudy image approach is already much better than the median-based approach. The greenest pixel mosaic provides reasonable results for both Singapore and Mumbai, but fails both for Jakarta and Bogota.

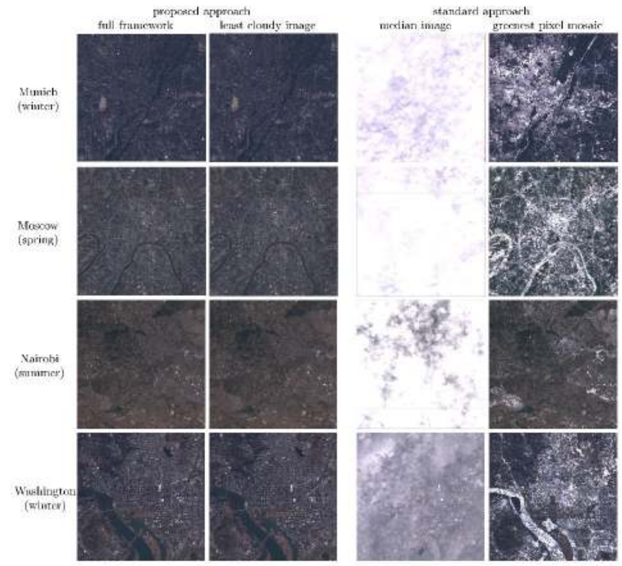


Figure 9. Cloud-free image aggregation examples for areas moderately affected by clouds.

3.4 Severely cloud-affected areas

Figure 11 compares the results achieved by our full approach and the least cloudy image based on the lowest bad pixel percentage (two leftmost columns) to the two standard approaches (two rightmost columns) for individual, often cloud-affected seasons of the cities of Guangzhou, Vancouver, Shenzhen, and Hongkong. Obviously, not even our approaches can deal with such situations, as there is not sufficient cloud-free information contained in the Sentinel-2 archive for the requested season. However, Figure 12 gives an impression what happens if just one extra month is added to the requested time period, thus giving the algorithm more chances to identify cloud-free input information. In this case, our full approach can make use of our least cloudy image and thus provide a useful cloud-free solution, while the median-based approach fails again. The greenest pixel mosaic approach is able to generate cloud-free images, but again introduces color artifacts for water surfaces and loses contrast in built up areas.

4. DISCUSSION

The results displayed in the previous section illustrate the general feasibility of the proposed approach for regions of interest, which are affected by cloud cover less than about 75% of the time. In these cases often already the least cloudy image provides a clean and artifact-free solution (cf. Figs. 8, 9, and 12). In this regard, it has to be noted that our sorting by cloud cover percentage is much more sophisticated than a sorting and selecting by the `cloudcoverpercentage` metadata tag provided with the original Sentinel-2 products: Since the `cloudcoverpercentage` value is calculated for the whole granule, it does not provide

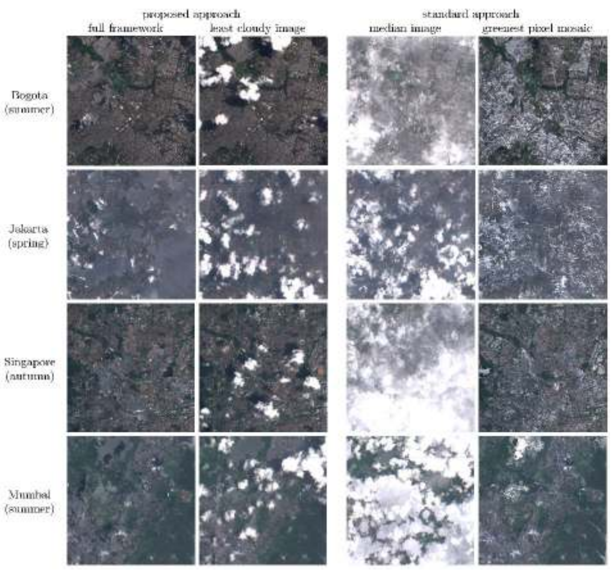


Figure 10. Cloud-free image aggregation examples for areas frequently affected by clouds.

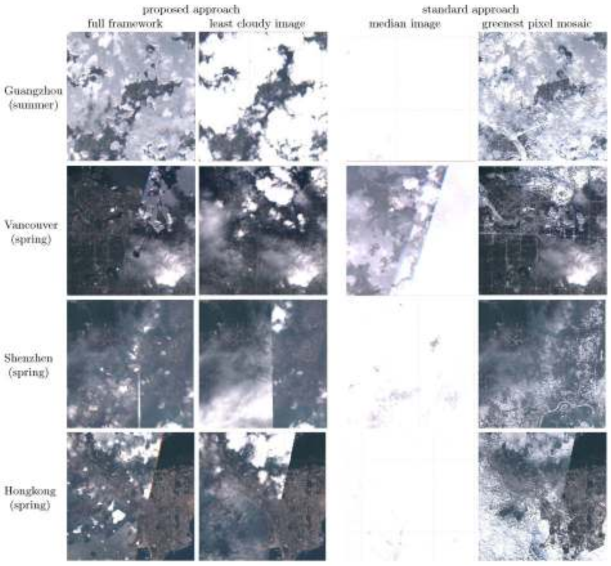


Figure 11. Cloud-free image aggregation examples for areas strongly affected by clouds.

useful information for the selection of images for specific regions of interest, which are smaller than a granule or covering more than one granule. Additionally, the `cloudcoverpercentage` is not directly related to the human perception of cloud cover and light cirrus clouds are counted towards the metric, even though most parts in the image are still largely visible with only mild haziness in some areas. As an example, the least cloudy image for Abuja, in Figure 8, has a `cloudcoverpercentage` of over 80% although it still provides useful information.

Furthermore, the results showed that our full framework, which aggregates an artificial image from multi-temporal data using the previously calculated quality score as prior information, can support the cloud-free image generation especially in areas where the cloud cover is so frequent that more than 5% of all pixels are affected by clouds even for the least cloudy image (cf. Figure 10).

The only case, in which all approaches fail, is for areas (and time frames) in which the cloud cover is persistent (i.e. more than about 75%). Then, there is simply not sufficient cloud-free information present in the multi-temporal data archive to recover a reasonable cloud-free image. The only solution in this case is to extend the time frame (cf. Figure 12), or to resort to data fusion-based machine learning approaches such as proposed by (Grohnfeldt et al., 2018).

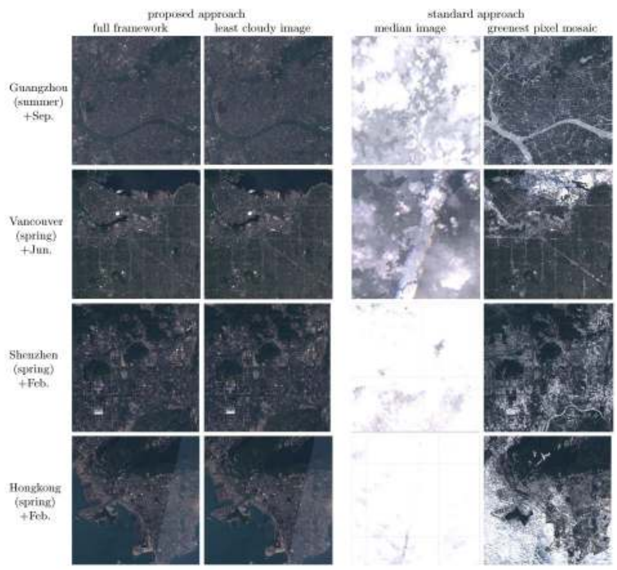


Figure 12. Cloud-free image aggregation examples for areas strongly affected by clouds, but with an extra month added to the seasonal (three-month) aggregation period.

5. SUMMARY AND CONCLUSION

With this paper, we have proposed an algorithm for the generation of cloud-free Sentinel-2 images for concise time periods and user-defined areas of interest using Google Earth Engine. By calculating a quality score, which is based on a pixel-wise analysis of cloud and shadow cover, it chooses either the least cloudy image as output or aggregates a cloud-free image from multiple input images acquired in the specified time frame, considering the individual pixel qualities. With experiments for areas of interest situated in locations of different cloud cover, we were able to confirm the feasibility of our approach. In all cases, it performs better than two often-used standard approaches (median image aggregation and greenest pixel mosaic). Implemented in Google Earth Engine's Python API, our approach can thus be used to generate cloud-free images for arbitrary areas of interest and time frames, which are much shorter than a year. The resulting images can then be used for a temporally fine-grained monitoring of land cover, which is not hampered by cloud-based information gaps.

ACKNOWLEDGMENTS

This work is jointly supported by the Helmholtz Association under the framework of the Young Investigators Group SiPEO (grant VH-NG-1018), the German Research Foundation (DFG, grant SCHM 3322/1-1), and the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement ERC-2016-StG-714087, Acronym: So2Sat).

REFERENCES

Candra, D.S., Phinn, S., Scarth, P., 2017. Cloud and cloud shadow removal of landsat 8 images using multitemporal cloud removal method. In: *2017 6th International Conference on Agro-Geoinformatics*, 1–5.

Chang, N.B., Bai, K., Chen, C.F., 2015. Smart information reconstruction via time-space-spectrum continuum for cloud removal in satellite images. *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.*,18(5), 1898–1912.

Cheng, Q., Shen, H., Zhang, L., Yuan, Q., Zeng, C., 2014. Cloud removal for remotely sensed images by similar pixel replacement guided with a spatio-temporal MRF model. *ISPRS J. Photogramm. Remote Sens.*,92, 54–68.

Cihlar, J., Howarth, J., 1994. Detection and removal of cloud contamination from AVHRR images. *IEEE Trans. Geosci. Remote Sens.*,32(3), 583–589.

European Space Agency, 2019. S2 MPC L1C Data Quality Report. Ref. S2-PDGS-MPC-DQR, Issue: 40. Technical report, ESA. Date: 6 June 2019.

Gao, B., 1996. NDWI – a normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sensing of Environment*,58(3), 257 – 266.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., 2017. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment* ,202, 18–27.

Grohnfeldt, C., Schmitt, M., Zhu, X.X., 2018. A conditional generative adversarial network to fuse SAR and multispectral optical data for cloud removal from Sentinel-2 images. In: *Proceedings of IEEE International Geoscience and Remote Sensing Symposium*,1726–1729.

Hall, D.K., Riggs, G. A., 2011. *Normalized-Difference Snow Index (NDSI)*. Springer Netherlands, Dordrecht, 779–780.

Huang, B., Li, Y., Han, X., Cui, Y., Li, W., Li, R., 2015. Cloud removal from optical satellite imagery with SAR imagery using sparse representation. *IEEE Geosci. Remote Sens. Lett.*,12(5), 1046–1050.

King, M.D., Platnick, S., Menzel, W.P., Ackerman, S.A., Hubanks, P.A., 2013. Spatial and temporal distribution of clouds observed by MODIS onboard the Terra and Aqua satellites. *IEEE Trans. Geosci. Remote Sens.*,51(7), 3826–3852.

Lin, C.H., Tsai, P.H., Lai, K.H., Chen, J.Y., 2013. Cloud removal from multitemporal satellite images using information cloning. *IEEE Trans. Geosci. Remote Sens.*,51(1), 232–241.

Qiu, C., Mou, L., Schmitt, M., Zhu, X.X., 2019. LCZ-based urban land cover classification from multi-seasonal Sentinel-2 images with a recurrent residual network. *ISPRS J. Photogramm. Remote Sens.*,154, 151–162.

Ramoino, F., Tutunaru, F., Pera, F., Arino, O., 2017. Ten-meter Sentinel-2A cloud-free composite – Southern Africa 2016. *Remote Sensing*,9(7), pp. 1–9.

Sentinel-2 cloudless, 2017. https://s2maps.eu/.

Xu, M., Jia, X., Pickering, M., Plaza, A.J., 2016. Cloud removal based on sparse representation via multitemporal dictionary learning. *IEEE Trans. Geosci. Remote Sens.*,54(5), 2998–3006.

Zhu, X., Gao, F., Liu, D., Chen, J., 2012. A modified neighborhood similar pixel interpolator approach for removing thick clouds in landsat images. *IEEE Geosci. Remote Sens. Lett.*,9(3), 521–525.