New tool for spatio-temporal image fusion in remote sensing: a case study approach using Sentinel-2 and Sentinel-3 data

Nikolina Mileva, Susanne Mecklenburg, Ferran Gascon

Angaben zur Veröffentlichung / Publication details:

Mileva, Nikolina, Susanne Mecklenburg, and Ferran Gascon. 2018. “New tool for spatio-temporal image fusion in remote sensing: a case study approach using Sentinel-2 and Sentinel-3 data.” In *Image and Signal Processing for Remote Sensing XXIV, 10-13 September 2018, Berlin, Germany*, edited by Lorenzo Bruzzone and Francesca Bovolo, 107890L. Bellingham, WA: SPIE. https://doi.org/10.1117/12.2327091.

New tool for spatio-temporal image fusion in remote sensing: a case study approach using Sentinel-2 and Sentinel-3 data

Nikolina Mileva, Susanne Mecklenburg, Ferran Gascon

Nikolina Mileva, Susanne Mecklenburg, Ferran Gascon, "New tool for spatio-temporal image fusion in remote sensing: a case study approach using Sentinel-2 and Sentinel-3 data," Proc. SPIE 10789, Image and Signal Processing for Remote Sensing XXIV, 107890L (9 October 2018); doi: 10.1117/12.2327091

Event: SPIE Remote Sensing, 2018, Berlin, Germany

New tool for spatiotemporal image fusion in remote sensing - a case study approach using Sentinel-2 and Sentinel-3 data

Nikolina Mileva[a], Susanne Mecklenburg[a], and Ferran Gascon[a]

[a]European Space Agency, 00044 Frascati (RM), Italy

ABSTRACT

Remote sensing image fusion allows the spectral, spatial and temporal enhancement of images. New techniques for image fusion are constantly emerging shifting the focus from pan-sharpening to spatiotemporal fusion of data originating from different sensors and platforms. However, the application of image fusion in the field of Earth observation still remains limited. The number and complexity of the different techniques available today can be overwhelming thus preventing users from fully exploiting the potential of fusion. The aim of this study is to make fusion products more accessible to users by providing them with a simple tool for spatiotemporal fusion in Python. This tool will contribute to the better exploitation of data from available sensors making possible to bring the images to the spectral, spatial and temporal resolution required by the user. The fusion algorithm implemented in the tool is based on the spatial and temporal adaptive reflectance fusion model (STARFM) - a well established fusion technique in the field of remote sensing often used as benchmark by other algorithms. The capabilities of the tool are demonstrated by three case studies using Sentinel-2 and simulated Sentinel-3 data. The first case study is about deforestation in the Amazon forest. The other two case studies concentrate on detecting change in an agricultural site in Southern Germany and urban flooding caused by the hurricane Harvey.

Keywords: Data fusion, Sentinel-2, Sentinel-3, remote sensing, surface reflectance

1. INTRODUCTION

The current constellation of Sentinel-2 allows the systematic acquisition of data over continental land surfaces and selected water surfaces every 5 days. The Multi-Spectral Instrument (MSI) on board of Sentinel-2 has 13 spectral bands ranging from the visible and near infrared to the short-wavelength infrared part of the electromagnetic spectrum with varying spatial resolution (10 m, 20 m, 60 m). However, the availability of clouds can lower the amount of optical data available for the purposes of change detection activities such as agricultural monitoring, deforestation and emergency management. A solution to this problem would be the use of an Earth observation sensor with better temporal resolution. The Ocean and Land Colour Instrument (OLCI) on board of the Sentinel-3 constellation has global coverage with revisit time of less than two days. OLCI has 21 spectral bands covering the visible and near-infrared part of the electromagnetic spectrum. However, the spatial resolution of the instrument is lower - 300 m, which can be an impeding factor for certain applications. Combining the high spatial resolution of MSI (Sentinel-2) with the frequent revisit time of OLCI (Sentinel-3) in a synthetic product could mitigate the trade-off between the spatial and temporal resolution of the two instruments.

Due to the limited length of the Sentinel-2 and Sentinel-3 time series there has been little research focused on the fusion of data from both platforms. Inglada et al. [1] compared different methods for the fusion of simulated Sentinel-2 and Sentinel-3 data - naive fusion by spatial interpolation, Savitzky-Golay interpolation, spatial and temporal adaptive reflectance fusion model (STARFM), Bayesian data fusion, sigmoid fitting, and a combination of the last two. STARFM had the best performance, however it has to be noted that the comparison was made based on land cover classification accuracy and no quality assessment was performed on the fused images. Mitraka and Berger [2] explored the synergistic use of Sentinel-2 and Sentinel-3 thermal bands for the estimation

Further author information: (Send correspondence to Nikolina Mileva)
Nikolina Mileva: E-mail: nikolina.mileva@esa.int, Telephone: +39 06 941 88515
Susanne Mecklenburg: E-mail: susanne.mecklenburg@esa.int, Telephone: +39 06 941 80695
Ferran Gascon: E-mail: ferran.gascon@esa.int, Telephone: +39 06 941 88605

Image and Signal Processing for Remote Sensing XXIV, edited by Lorenzo Bruzzone,
Francesca Bovolo, Proc. of SPIE Vol. 10789, 107890L · © 2018 SPIE
CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2327091

of land surface temperature at 20 m resolution using spatial-spectral unmixing. They also used simulated data derived from ground-based and airborne measurements. Several researchers proposed fusion models applicable to Sentinel-2/Sentinel-3 but due to the lack of data either from both or from one of the two sensors used similar sensor pairs such as SPOT4/MODIS [3] and MSI/MODIS [4],[5]. Doxani et al. [3] used SPOT4 as a substitute for MSI and MODIS as a substitute for OLCI in a new fusion approach based on spectral unmixing. Similarly, Korosov and Pozdnyakov [4] applied a neural network model for the fusion of MSI and MODIS. Wu et al. [5] used fusion of MODIS and Sentinel-2 time series to create a synthetic NDVI product at 10 m resolution. They applied an improved version of the spatial and temporal data fusion approach (ISTDFA) originally developed for Landsat and MODIS [6]. Recently, Wang and Atkinson [7] proposed a new fusion method Fit-FC specifically tailored towards the fusion of Sentinel-2 MSI and Sentinel-3 OLCI data. Their approach relies on a linear regression model trained on a pair of OLCI images. Similarly to STARFM [8], the resulting fusion product is further improved by applying a weighting function based on the spectral similarity of the neighboring pixels and the spatial distance to the predicted pixel. It has to be noted that they have simulated the Sentinel-3 images by using resampled Sentinel-2 images thus geolocation errors and errors coming from the differences in atmospheric correction are excluded.

The blending of Landsat and MODIS images on the other hand has been well researched and offers a good foundation for the development of a fusion workflow for Sentinel-2 and Sentinel-3 data. One of the prerequisites making possible the fusion of Landsat and MODIS is the similarity of their orbital parameters [8]. Such similarity can be observed also between the Sentinel-2 and Sentinel-3 satellites (Table 1).

Table 1. Orbit inclination and Mean Local Solar Time (MLST) of selected satellites

Orbital parameters	Landsat-7	Landsat-8	Terra MODIS	Sentinel-2	Sentinel-3
Orbit inclination	98.2°	98.2°	98.5°	98.6°	98.65°
MLST	10:00±15 min	10:11±15 min	10:30	10:30	10:00

Similarly to MODIS, OLCI has narrower bands than MSI with one or more bands falling within the wavelength range of the respective MSI band (Table 2).

Table 2. Sentinel-2 MSI and Sentinel-3 OLCI selected bandwidths

	Sentinel-2 MSI			Sentinel-3 OLCI		
(nm)	Band name	λ center	Width	Band name	λ center	Width
Blue	B2	490	65	Oa04, Oa05	490, 510	10
Green	B3	560	35	Oa06	560	10
Red	B4	665	30	Oa08	665	10
NIR (broad)	B8	842	115	Oa14, Oa15, Oa16 Oa17, Oa18, Oa19	764, 768, 779 865, 885, 900	3.75, 2.5, 15 20, 10, 10
NIR (narrow)	B8A	865	20	Oa13	865	20

In remote sensing, image fusion can be used either for gap filling of time series where images are available both before and after the day of prediction, or purely for prediction where training data is available only before or after the day of prediction. In this study, the second case is considered.

2. METHODS

Gao et al. [8] proposed a spatial and temporal adaptive reflectance fusion model (STARFM) for fusing Landsat and MODIS surface reflectance with the aim to generate a synthetic Landsat-like product at the same temporal resolution as MODIS. Their model is able to retrieve fine resolution surface reflectance from one or two pairs of Landsat/MODIS images used for training and an additional MODIS image at the date of prediction. In order to determine the surface reflectance at fine resolution, information from neighboring pixels is used. These neighboring pixels have to be spectrally similar and homogeneous. Additional weights are applied based on the spectral and temporal differences between the Landsat and MODIS images and depending on the spatial proximity of the neighboring pixels to the predicted pixel.

In this study, a Python implementation of STARFM is suggested. The original implementation of STARFM is in C language. To showcase the abilities of the tool Sentinel-2 and Sentinel-3 (OLCI) data is used. In order to avoid errors from differences in the atmospheric correction, geolocation errors and other artifacts caused by preprocessing steps such as collocation and resampling, the OLCI images are simulated by downscaling Sentinel-2 images to 300 m resolution. As input the atmospherically corrected Level 2A product of Sentinel-2 is used over Europe. It is freely available on the Copernicus Open Access Hub (https://scihub.copernicus.eu/). For test sites outside of Europe the atmospheric correction is performed using the Sen2Cor plug-in in SNAP.

2.1 Theoretical basis

In STARFM [8], it is assumed that the fine resolution surface reflectance (F) at the date of prediction (t_1) can be retrieved by using information from a fine resolution image (F) at a date preceding/following the date of prediction (t_0), a coarse resolution image (C) at the same date (t_0), and a coarse resolution image (C) at the date of prediction (t_1). If (x_i, y_j) determine the pixel location and w is the size of the moving window within which the search for similar pixels is performed, the fine resolution surface reflectance at t_1 can be calculated as follow:

$$F(x_{w/2}, y_{w/2}, t_1) = \sum_{i=1}^{w} \sum_{j=1}^{w} \sum_{k=1}^{n} W_{ijk} \times \left(C(x_i, y_j, t_1) + F(x_i, y_j, t_0) - C(x_i, y_j, t_0) \right), \tag{1}$$

where W_{ijk} is a weighting parameter comprised by the spectral (2), temporal (3) and spatial (4) distances. These are defined as follow:

$$S_{ijk} = |F(x_i, y_j, t_0) - C(x_i, y_j, t_0)| + 1\,, \tag{2}$$

where S is the spectral distance between the fine resolution image F and the coarse resolution image C at the date on which the training images are available (t_0). A 1 is added in order to avoid zero values;

$$T_{ijk} = |C(x_i, y_j, t_0) - C(x_i, y_j, t_1)| + 1\,, \tag{3}$$

where T is the temporal distance between the two coarse resolution images at date t_0 and t_1. Again, a 1 is added in order to avoid zero values;

$$d_{ijk} = \sqrt{(x_{w/2} - x_i)^2 + (y_{w/2} - y_j)^2}\,, \tag{4}$$

where d is the spatial distance between the central predicted pixel $(x_{w/2}, y_{w/2})$ and the neighboring pixel (x_i, y_j). The spatial distance is converted to relative distance:

$$D_{ijk} = \frac{d_{ijk}}{A} + 1\,, \tag{5}$$

where A is a constant defining the relative importance of the spatial distance to the other weighting parameters.

The combined distance is calculated by taking the inverse of the different distances, so the neighboring pixels which have smaller spectral and temporal distance and are closer to the central pixel get larger weight:

$$C_{ijk} = \frac{1}{S_{ijk}} \times \frac{1}{T_{ijk}} \times \frac{1}{D_{ijk}} \tag{6}$$

Alternatively, the natural logarithm of the weighting distances can be used in order to decrease the sensitivity to the spectral distance:

$$C_{ijk} = \frac{1}{ln(S_{ijk}+1)} \times \frac{1}{ln(T_{ijk}+1)} \times \frac{1}{ln(D_{ijk}+1)} \tag{7}$$

The final weighting parameter is calculated by normalizing the combined distance so that the sum of the weights is equal to one:

$$W_{ijk} = \frac{C_{ijk}}{\sum_{i=1}^{w}\sum_{j=1}^{w}\sum_{k=1}^{n} C_{ijk}} \tag{8}$$

Weights are applied only on similar pixels. These are defined during a "dynamic classification process"[9]. Within each moving window a threshold (9) is defined and depending on whether the pixel value is below or above this threshold, it is classified as similar or not similar [10]. The threshold is defined as follow:

$$|F(x_i, y_j, t_0) - F(x_{w/2}, y_{w/2}, t_0)| \leq 2\sigma/m\,, \tag{9}$$

where σ is the standard deviation and m is the number of classes. Additionally, the similar pixels are also filtered based on two criteria:

$$S_{ijk} < max(|F(x_{w/2}, y_{w/2}, t_0) - M(x_{w/2}, y_{w/2}, t_0)|) + \sigma_{fc} \tag{10}$$

and

$$T_{ijk} < max(|M(x_{w/2}, y_{w/2}, t_0) - M(x_{w/2}, y_{w/2}, t_1)|) + \sigma_{cc}\,, \tag{11}$$

where σ_{fc} is the uncertainty for the spectral distance (2), and σ_{cc} is the uncertainty for the temporal distance (3). These uncertainties are derived from the uncertainty of the fine resolution sensor (σ_f) and of the coarse resolution sensor (σ_c):

$$\sigma_{fc} = \sqrt{\sigma_f^2 + \sigma_c^2}\,, \tag{12}$$

$$\sigma_{cc} = \sqrt{\sigma_c^2 + \sigma_c^2} = \sqrt{2}\sigma_c^2\,. \tag{13}$$

2.2 Implications for the Python implementation

The first problem that has to be solved when implementing STARFM in Python is performing the calculations within the moving window - the most time consuming part of the algorithm [11]. While using loops in low level languages such as C is efficient, such operations are significantly slower in Python. Using a vectorised solution can speed up the process. However, it requires the calculations to be performed in the memory of the computer. Thus, running the algorithm over larger areas (such as one Sentinel-2 tile) would result in running out of memory. In the current implementation, the moving window operations were implemented using generators. Generators in Python create iterator objects which require less memory. Such implementation is still relatively slow. But improvements are possible by using parallelization solutions such as Dask [12] which allow to fully exploit the resources of a single machine by leveraging between the available memory and disk space.

In STARFM, several different operations are performed within the moving window such as calculating the standard deviation among all pixels within the window in order to set a similarity threshold. In the initial version of STARFM, the similar pixels are determined separately for each band [9]. An enhanced version of STARFM (ESTARFM) [10] suggests the use of all bands for determining the spectrally similar pixels. In the first case, the calculations have to be performed in a 2D space but including information from the other bands will increase the dimensions involved. Such operations can be challenging in Python. So, in order to decrease the dimensionality of the problem each window is flattened into a row (Fig. 1a). Thus, the dimensions involved are decreased by a factor of two. The drawback of such an approach is the creation of redundant data.

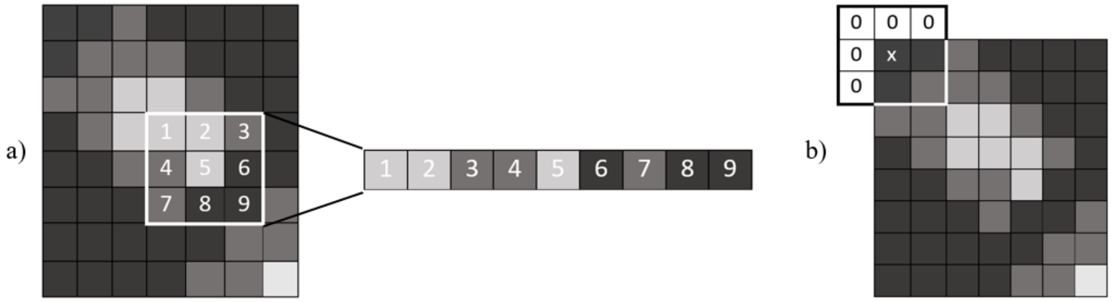


Figure 1. The pixels within each moving window are flattened into a row in order to mitigate dimensionality issues (a). Padding with zeros the borders of an image (b).

Another important issue is the handling of the pixels at the edges of the image. As each pixel has to be in the center of the moving window once, the moving window has to be forced to go over pixels for which the neighboring pixels are partially missing (i.e. at the edges of the image). One way to solve this issue is to use convolution which can neatly deal with such cases. This was the approach used to calculate the standard deviation within the moving window. However, not every operation can be easily represented as convolution. For this purpose, the input images were padded with zeros (Fig. 1b). It has to be noted that border effects are inevitable as for the borders of the image there will be always less pixels taken into consideration than in the center of the image. Increasing the size of the moving window can reinforce this effect.

3. RESULTS

3.1 Tests with simulated data

In order to verify that the Python implementation is consistent with the original version of STARFM, several tests were performed using the same simulated data as in Gao et al. [8]. STARFM can be used with both one or two pairs of fine/coarse resolution images. For the purposes of the tests only the first case (with one pair of training images) was considered. All tests were performed using a sliding window of 51 pixels (fine resolution) corresponding approximately to 3 coarse resolution pixels, uncertainty of 0.005 ($\sigma_{fc} = \sigma_{cc}$), and spatial impact factor of 250.

The first test case (Fig. 2) is used to evaluate the ability of the algorithm to retrieve changing surface reflectance. In the test image, two different classes are introduced ($m = 2$) - water and vegetation. The surface reflectance of water does not change over time and both at date t_0 and t_1 is equal to 0.05. The surface reflectance of vegetation changes from 0.1 at t_0 to 0.2 at t_1. The shapes of the water and vegetation areas do not change over time. The algorithm is able to detect the change in surface reflectance for the vegetated area with some small inaccuracies around the edges of the water body. This result is consistent with Gao et al. [8].

In the second test case (Fig. 3), the ability of the algorithm to detect changing shapes is assessed. The surface reflectance for both object classes is constant while the shape for the circular object (water) is increasing from 500 m to 2000 m. The results show that the surface reflectance could not be properly predicted for the boundary of the water body at its final state.

The third test case (Fig. 4) evaluates if the algorithm is able to detect linear features (e.g. roads, bridges) having both static (water) and changing (vegetation) backgrounds. Similarly to Gao et al. [8], the linear feature is retrieved correctly when it is surrounded by stable environment (e.g. water) and some inaccuracies are observed when the background is changing (e.g. changing vegetation).

3.2 Tests with Sentinel-2 data

First, the uncertainty (σ) is set to 3% which corresponds to the absolute radiometric accuracy of the MSI sensor [13]. The size of the moving window is 31 pixels (310 m). The number of land cover classes used is 4. The spatial impact factor (A) is set to 150 which corresponds approximately to half the size of the window.

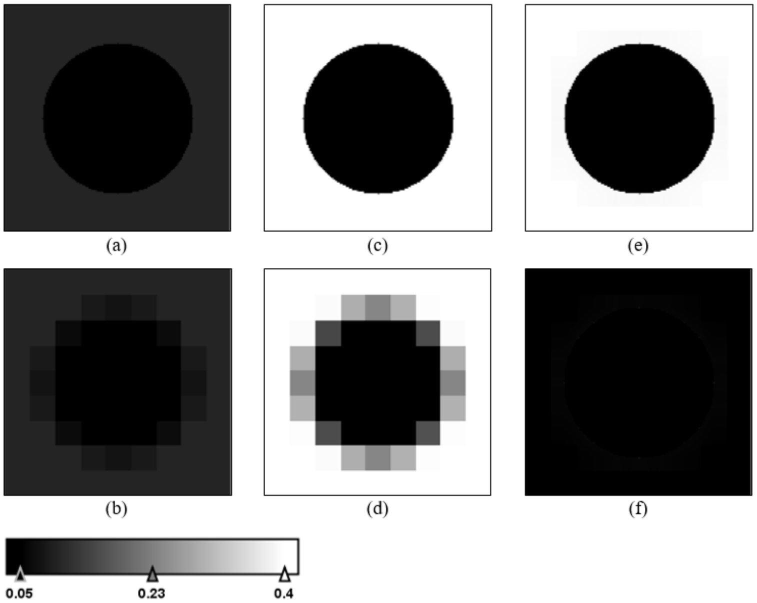


Figure 2. Test of STARFM with one pair of training images at t_0 - (a) and (b) representing respectively the fine and the coarse resolution images, and one coarse resolution image (d) at t_1. The output of the prediction (e) is compared to the actual fine resolution image at t_1 (c). The difference between the two is shown in (f). In this simulated dataset, water is represented as a circle with constant surface reflectance of 0.05 and the background is vegetation with changing surface reflectance from 0.1 to 0.4.

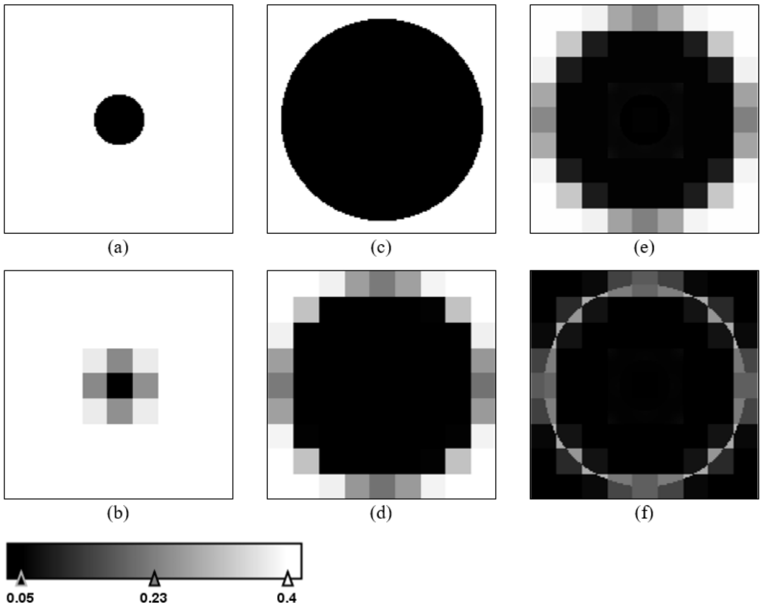


Figure 3. Test of STARFM with one pair of training images at t_0 - (a) and (b) representing respectively the fine and the coarse resolution images, and one coarse resolution image (d) at t_1. The output of the prediction (e) is compared to the actual fine resolution image at t_1 (c). The difference between the two is shown in (f). In this simulated dataset, the circle represents a feature with a changing shape. The surface reflectance is remaining constant.

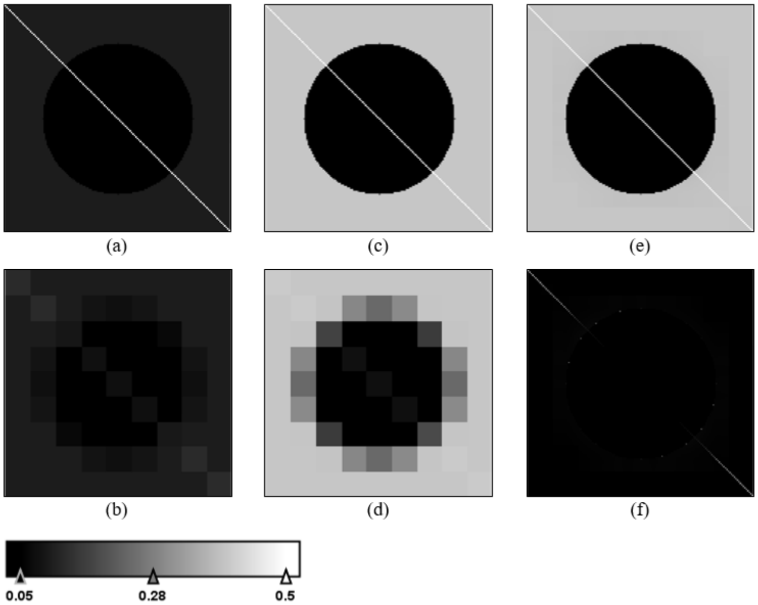


Figure 4. Test of STARFM with one pair of training images at t_0 - (a) and (b) representing respectively the fine and the coarse resolution images, and one coarse resolution image (d) at t_1. The output of the prediction (e) is compared to the actual fine resolution image at t_1 (c). The difference between the two is shown in (f). In this simulated dataset, the line represents a linear feature (road/bridge) with constant reflectance and shape. The vegetation background has changing surface reflectance while the surface reflectance of the water body is remaining constant.

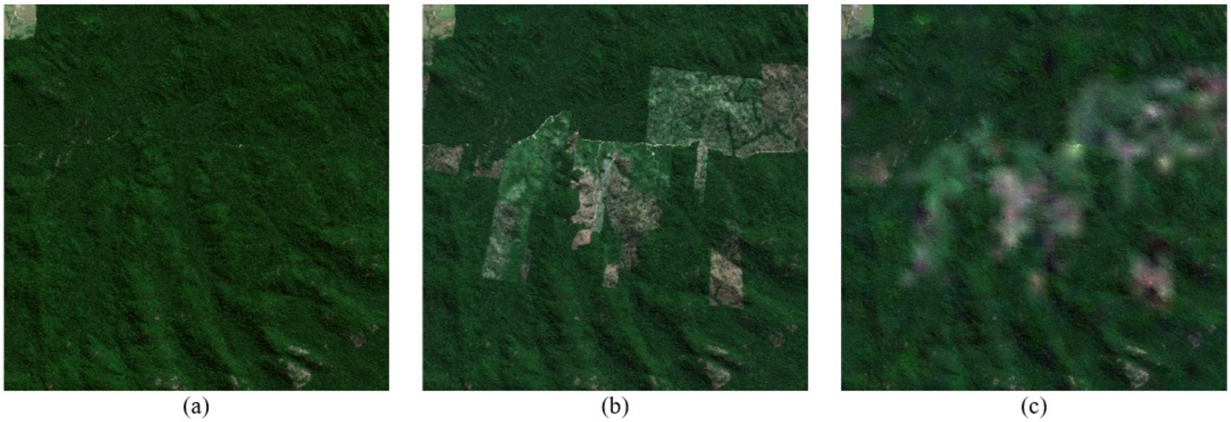


Figure 5. The result of fusion with STARFM over a deforested area in the Amazon forest (c) and the corresponding Sentinel-2 images from 12.07.2017 (a) and 12.07.2018 (b).

The first case study is about detecting deforestation in the Amazon forest. The study area is a 9x9 km plot near Ji-Parana, Rondonia in Brazil (10°39′38″ S, 61°20′10″ W). Two Sentinel-2 images are used as input - from 12.07.2017 and from 12.07.2018 (Fig. 5). The two images are resampled to 300 m resolution to simulate Sentinel-3 OLCI images. The results (Fig. 5c) prove that STARFM is able to detect the changes in forest cover. Comparing the differences in surface reflectance that occurred between t_0 and t_1 with the differences between the predicted and the actual surface reflectance at t_1 (Fig. 6) shows that additional information is acquired from the coarse resolution image.

The second case study is about detecting changes in agricultural fields. The test site is located near Augsburg, Germany (48°13′18″ N, 10°56′51″ E). As input two Sentinel-2 images from 22.04.2018 and 07.05.2018 are used

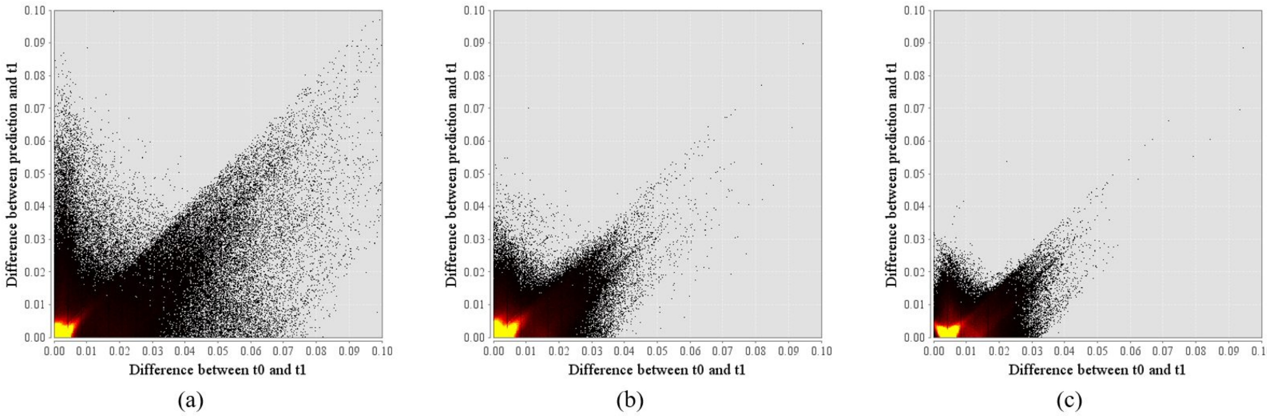


Figure 6. The differences between the actual and the predicted surface reflectance are smaller than the differences between the surface reflectance on 12.07.2017 (t_0) and 12.07.2018 (t_1) for the red (a), green (b) and blue bands (c) showing that the fusion approach is successful at capturing change.

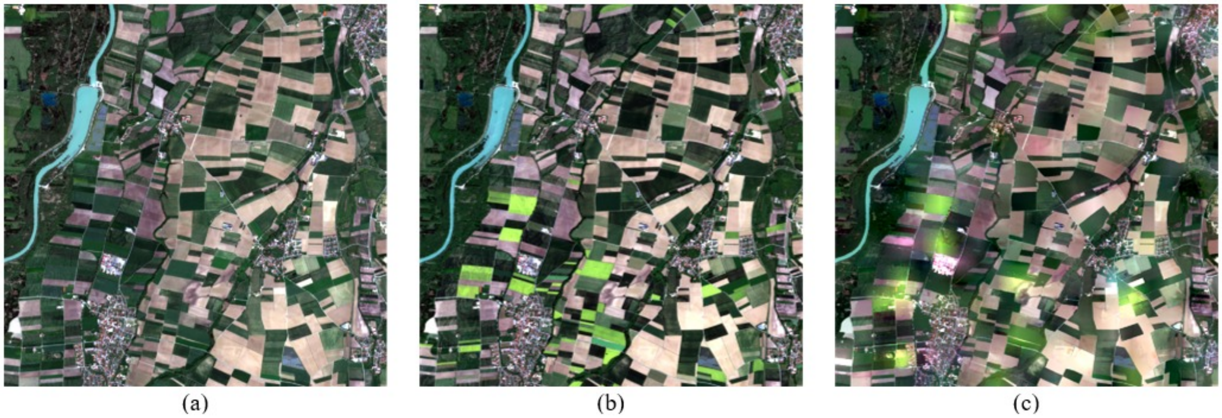
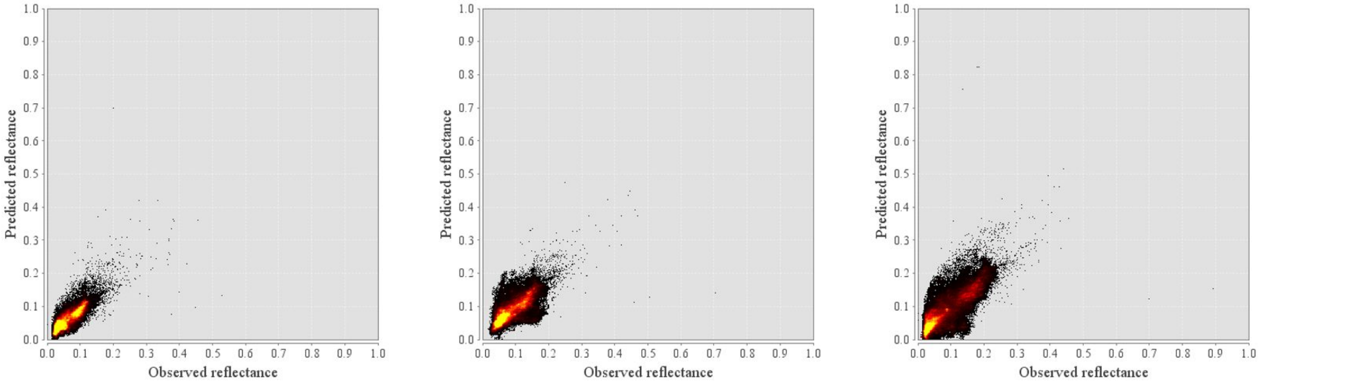


Figure 7. The result of fusion with STARFM over an agricultural area (c) and the corresponding Sentinel-2 images from 22.04.2018 (a) and 07.05.2018 (b).

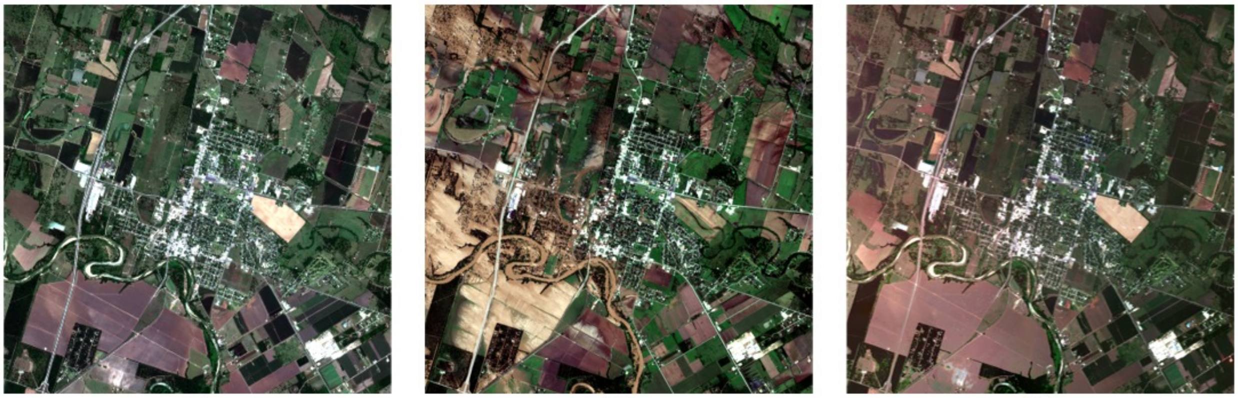
showing the rapid changes during the spring period (Fig. 7). The 6x6 km test site is mostly covered by agricultural land but there are also roads, houses, woodland and a river making the scene even more complex. Because of the increased complexity of the region, the logarithmic weighting function is applied (7). Comparing the observed and the predicted surface reflectance shows that there is a satisfactory overlap between the two as evidenced by their linear relationship (Fig. 8). However, for fields with size smaller than the pixel size of the coarse resolution sensor detecting change is difficult.

The third case study explores the possibility to map floods by using fusion (Fig. 9). The study area is located in Wharton, Texas in the USA (29°19′16″ N, 96°05′47″ W) - a city affected by the hurricane Harvey. A Sentinel-2 MSI image from 30.08.2017 shows the damage caused by the hurricane, and an image from 17.05.2018 is used as reference due to the lack of cloud free images prior the disaster. The 9x9 km plot includes both some urban and agricultural areas. Due to the complexity of the region the size of the moving window is increased to 910 m (91 fine resolution pixels) and the spatial impact factor is set to 37.5. A logarithmic weighting function is applied (7). The results show that over larger areas the differences in surface reflectance can be identified (Fig. 10). However, for urban areas the detection of change is challenging also because the resolution of 10 m is not always sufficient to capture urban objects (e.g. houses) in such detail as it is needed for urban flood mapping.



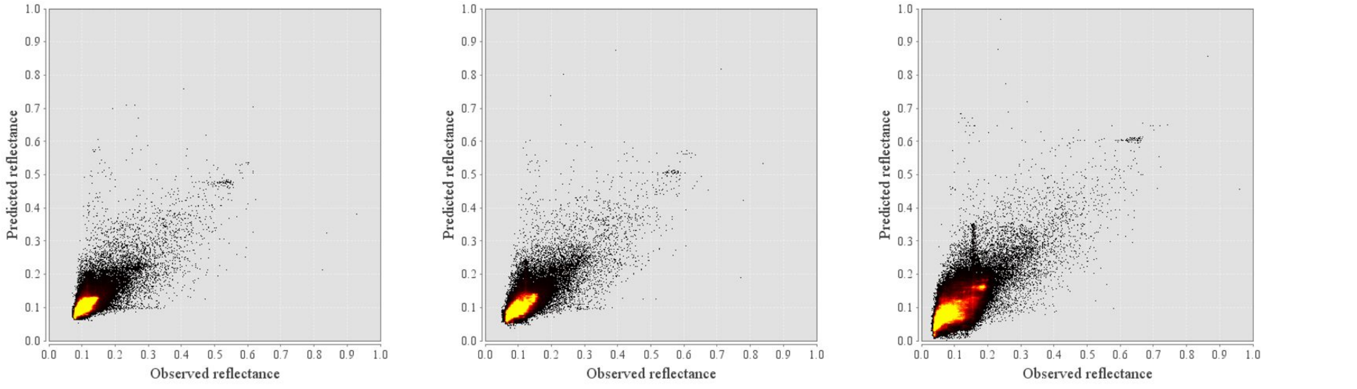
(a) (b) (c)

Figure 8. Comparison between the observed and predicted surface reflectance for the red (a), green (b) and blue (c) bands for the agricultural monitoring scenario.



(a) (b) (c)

Figure 9. The result of fusion with STARFM over a flooded area (c) and the corresponding Sentinel-2 images from 17.05.2018 (a) and 30.08.2017 (b).



(a) (b) (c)

Figure 10. Comparison between the observed and predicted surface reflectance for the red (a), green (b) and blue (c) bands for the flooding scenario.

4. CONCLUSION AND DISCUSSION

This study described the implementation of STARFM in Python and showed some applications of fusion using Sentinel-2 data. For the three scenarios considered - deforestation, agricultural monitoring and urban flooding, one pair of fine/coarse resolution images was used together with one coarse resolution image at the day of prediction.

The result of fusion with STARFM largely depends on the geographical homogeneity of the area of interest. STARFM provides best results when pure coarse resolution pixels are available (e.g. deforestation scenario). However, agricultural fields are often smaller in size than the pixel size of the coarse resolution sensor. In such cases, Gao et al. [8] suggest the use of a bigger searching window with a weighting function less sensitive to the spectral distance (7) and more sensitive to the spatial distance. But for complex regions with many different land use classes such as agricultural and urban areas, fusion can still be challenging.

Here, STARFM was applied on rather small test sites ranging from 36 km^2 to 81 km^2. Applying the algorithm on a larger area (e.g. one Sentinel-2 tile) could increase the number of similar pixels found and thus improve the result.

Another important aspect for the application of STARFM is the assumption that the surface reflectance provided by the fine and coarse resolution sensors is linearly correlated. While this assumption is fulfilled for Landsat and MODIS data, it might not be true for other sensors [10]. Thus, the use of STARFM with real OLCI/MSI data might require additional bandpass adjustments to make the measurements from both sensors comparable. It is also important to apply a similar atmospheric correction method as this was done for MODIS and Landsat [8].

REFERENCES

[1] Inglada, J., Osman, J., and Yin, T., "Fusion of Sentinel-2 and PROBA-V / Sentinel-3 Images for Multi-Temporal Land-Cover Map Production," in [*32nd EARSeL Symposium Proceedings "Advances in Geosciences"*], 475–483 (2012).

[2] Mitraka, Z. and Berger, M., "Land surface emissivity estimation from synergistic use of Sentinel-2 and Sentinel-3 data based on spatial-spectral unmixing techniques," *European Space Agency, (Special Publication) ESA SP* **707** (2012).

[3] Doxani, G., Mitraka, Z., Gascon, F., Goryl, P., and Bojkov, B., "A spectral unmixing model for the integration of multi-sensor imagery: A tool to generate consistent time series data," *Remote Sensing* **7**(10), 14000–14018 (2015).

[4] Korosov, A. and Pozdnyakov, D., "Fusion of data from Sentinel-2/MSI and Sentinel-3/OLCI," *European Space Agency, (Special Publication) ESA SP* **740** (2016).

[5] Wu, M., Yang, C., Song, X., Hoffmann, W., Huang, W., Niu, Z., Wang, C., Li, W., and Yu, B., "Monitoring cotton root rot by synthetic Sentinel-2 NDVI time series using improved spatial and temporal data fusion," *Scientific Reports* **8**(1) (2018).

[6] Wu, M., Wu, C., Huang, W., Niu, Z., Wang, C., Li, W., and Hao, P., "An improved high spatial and temporal data fusion approach for combining Landsat and MODIS data to generate daily synthetic Landsat imagery," *Information Fusion* **31**, 14–25 (2016).

[7] Wang, Q. and Atkinson, P., "Spatio-temporal fusion for daily Sentinel-2 images," *Remote Sensing of Environment* **204**, 31–42 (2018).

[8] Gao, F., Masek, J., Schwaller, M., and Hall, F., "On the blending of the Landsat and MODIS surface reflectance: Predicting daily Landsat surface reflectance," *IEEE Transactions on Geoscience and Remote Sensing* **44**(8), 2207–2218 (2006).

[9] Gao, F., Hilker, T., Zhu, X., Anderson, M., Masek, J., Wang, P., and Yang, Y., "Fusing Landsat and MODIS Data for Vegetation Monitoring," *IEEE Geoscience and Remote Sensing Magazine* **3**(3), 47–60 (2015).

[10] Zhu, X., Chen, J., Gao, F., Chen, X., and Masek, J., "An enhanced spatial and temporal adaptive reflectance fusion model for complex heterogeneous regions," *Remote Sensing of Environment* **114**(11), 2610–2623 (2010).

[11] Gao, F., Anderson, M., Zhang, X., Yang, Z., Alfieri, J., Kustas, W., Mueller, R., Johnson, D., and Prueger, J., "Toward mapping crop progress at field scales through fusion of Landsat and MODIS imagery," *Remote Sensing of Environment* **188**, 9–25 (2017).

[12] Dask Development Team, *Dask: Library for dynamic task scheduling* (2016).

[13] Drusch, M., Bello, U. D., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., Meygret, A., Spoto, F., Sy, O., Marchese, F., and Bargellini", P., "Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services," *Remote Sensing of Environment* **120**, 25–36 (2012).