User’s Guide to COSI-CORR Python code
Co-registration of Optically Sensed Images and Correlation
Correlation 2D Examples

Marie Cadoux, Saif Aati, Jean-Philippe Avouac

California Institute Of Technology

1200 East California Blvd, Pasadena, CA 91125, USA

July, 2023

Abstract

This document is a user’s guide to creating a 2D correlation map from examples. All the input parameters and results are present to help the user to understand the code. Three examples are treated to measure the ground surface deformations of a coseismic deformation (earthquake_ridgecrest example), an ice flow (glacier_jakobshavn example), and a sand dune migration (Saudi Arabia example). This code is also adapted to measure the ground surface deformation of a slow landslide.

Contents

1. Introduction

Correlation_2D_examples is a folder containing three examples of code allowing to get a 2D correlation map, in different fields of geology with differents kinds of image:

- dune_saudi_arabia (sand dune migration), Sentinel 2 images,
- earthquake_ridgecrest (coseismic deformation), NAIP images,
- glacier_jakobshavn (ice flow), Landsat 8 and 9 images.

With the 2D correlation map, COSI-Corr gets the horizontal ground surface deformation, East/West and North/South and the SNR (Signal Noise Ratio). These examples use the same code with different input parameters. For the three examples, all the input parameters and the results are present. The goal of this folder is to help the user with the first uses of the code.

2. Structure of the example

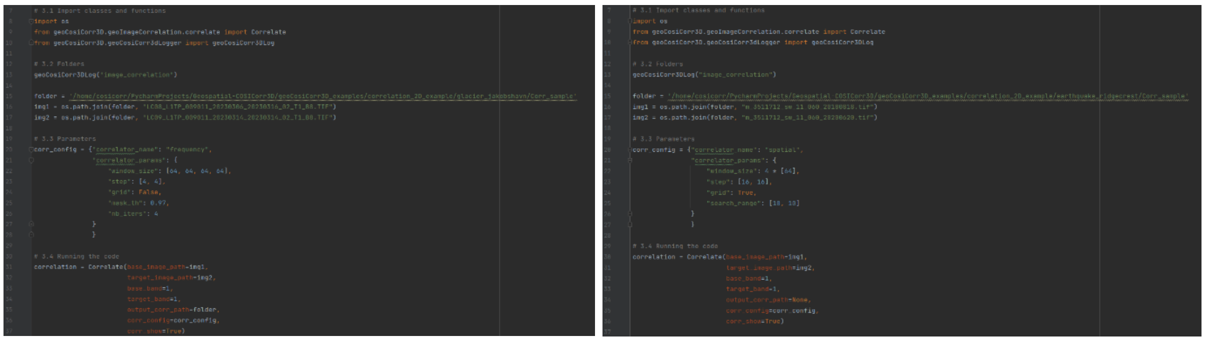
Correlation_2D_example contains three folders: dune_saudi_arabia, earthquake_ridgecrest and glacier_jakobshavn.

In each folder, there are two files:

- correlation.py: the code,
- corr_sample: contains the input parameters (the two images to be correlated) and a folder with the results.

3. Code operation

correlation.py is the code to get the 2D correlation map. Depending on the input parameter, the code changes: between frequency and spatial correlator (Figure 1). glacier_jakobshavn and dune_saudi_arabia use frequency correlator and earthquake_ridgecrest uses spatial correlator.



```python
# 3.1 Import classes and functions
import os
from geoCosiCorr3D.geoImageCorrelation.correlate import Correlate
from geoCosiCorr3D.geoCosiCorr3dLogger import geoCosiCorr3DLog

# 3.2 Folders
geoCosiCorr3DLog('image_correlation')

folder = '/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/correlation_2D_example/glacier_jakobshavn/Corr_sample'
img1 = os.path.join(folder, "LC08_L1TP_009011_20230306_20230316_02_T1_B8.TIF")
img2 = os.path.join(folder, "LC09_L1TP_009011_20230314_20230314_02_T1_B8.TIF")

# 3.3 Parameters
corr_config = {"correlator_name": "frequency",
               "correlator_params": {
                   "window_size": [64, 64, 64, 64],
                   "step": [4, 4],
                   "grid": False,
                   "mask_th": 0.97,
                   "nb_iters": 4
               }
               }

# 3.4 Running the code
correlation = Correlate(base_image_path=img1,
                        target_image_path=img2,
                        base_band=1,
                        target_band=1,
                        output_corr_path=folder,
                        corr_config=corr_config,
                        corr_show=True)
```

```python
# 3.1 Import classes and functions
import os
from geoCosiCorr3D.geoImageCorrelation.correlate import Correlate
from geoCosiCorr3D.geoCosiCorr3dLogger import geoCosiCorr3DLog

# 3.2 Folders
geoCosiCorr3DLog("image_correlation")

folder = '/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/correlation_2D_example/earthquake_ridgecrest/Corr_sample'
img1 = os.path.join(folder, "m_3511712_sw_11_060_20180818.tif")
img2 = os.path.join(folder, "m_3511712_sw_11_060_20200620.tif")

# 3.3 Parameters
corr_config = {"correlator_name": "spatial",
               "correlator_params": {
                   "window_size": 4 * [64],
                   "step": [16, 16],
                   "grid": True,
                   "search_range": [10, 10]
               }
               }

# 3.4 Running the code
correlation = Correlate(base_image_path=img1,
                        target_image_path=img2,
                        base_band=1,
                        target_band=1,
                        output_corr_path=None,
                        corr_config=corr_config,
                        corr_show=True)
```

Figure 1: Code to get the 2D correlation map with the frequency correlator (left) and the spatial correlator (right)

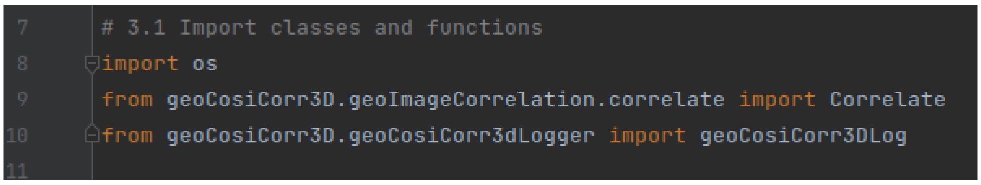
The code is organized into different parts:

- Import classes and functions
- Folders
- Parameters
- Running the code

We will explain these parts below.

3.1 Import classes and functions

The first part of the code consists of importing classes and functions to run correlate.py (Figure 2).



```python
# 3.1 Import classes and functions
import os
from geoCosiCorr3D.geoImageCorrelation.correlate import Correlate
from geoCosiCorr3D.geoCosiCorr3dLogger import geoCosiCorr3DLog
```

Figure 2: Import of classes and functions to run the correlation code

3.2 Folders

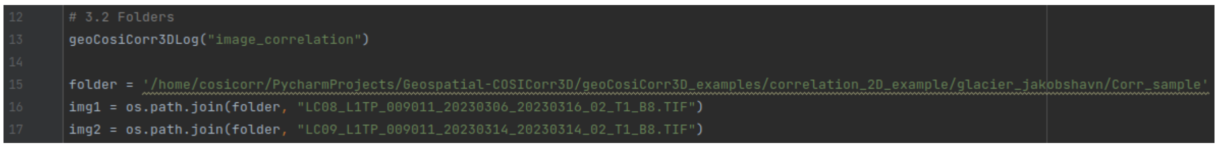
The second part of the code consists of writing the different paths of the input parameters and the file that contains the coding process (Figure 3).

1. **geoCosiCorr3Dlog("image_correlation"):** creates a file image_correlation_date_hour.log in GEO_COSI_CORR_3D_WD with information about the coding process. It is possible to change the name of this file by writing another message in quotation marks.

2. **Folder = '…' :** write the path to assign "Folder" to Corr_sample folder in the quotation marks.

3. **Img1 = os.path.join(folder, "m_3511712_sw_060_20180818.tif") :** assigns to Img1 the oldest image we want to correlate. It calls "base image". "m_3511712_sw_060_20180818.tif" is the name of the image. To correlate another image, we must write the name of the other image, located in Corr_sample.

4. **Img2 = os.path.join(folder, "m_3511712_sw_060_20200620.tif") :** assigns to Img2 the youngest image we want to correlate. It calls "target image". "m_3511712_sw_060_20200620.tif" is the name of the image. To correlate another image, we must write the name of the other image located in Corr_sample.



```python
# 3.2 Folders
geoCosiCorr3DLog("image_correlation")

folder = '/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/correlation_2D_example/glacier_jakobshavn/Corr_sample'
img1 = os.path.join(folder, "LC08_L1TP_009011_20230306_20230316_02_T1_B8.TIF")
img2 = os.path.join(folder, "LC09_L1TP_009011_20230314_20230314_02_T1_B8.TIF")
```

Figure 3: Paths to access images and create "image_correlation" file

3.3 Parameters

The third part of the code consists of writing the parameters of the correlation. By changing these parameters we will get different results. Depending on the correlator_name (frequency or spatial), the code is different (Figure 4). glacier_jakobshavn and dune_saudi_arabia use frequency correlator and earthquake_ridgecrest uses spatial correlator. We will explain these parameters to get the best 2D correlation map.

1. **"correlator_name":** write the correlation method used to correlate the images. Two correlators are available: "frequency" or "spatial".
   - **(a)** "frequency": the frequency correlator is Fourier based and is more accurate and faster than the spatial one. It should be used as a priority when correlating optical images and if we are processing a large data cube of images. However, this correlator is more sensitive to noise and is therefore recommended for optical images of good quality (Ayoub et al., 2017).
   - **(b)** "spatial": the spatial correlator maximizes the absolute value of the correlation coefficient and is coarser but more robust than the frequency one. Its use is recommended for correlating noisy optical images that provided bad results with the frequency correlator, or for correlating images of different content such as an optical image with a shaded DEM (Ayoub et al., 2017).

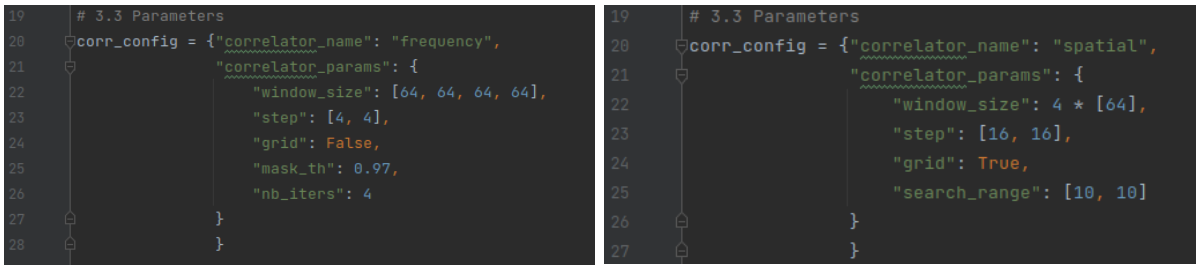
2. **"correlator_params":** write the different parameters of the correlation.
   - **(a) "window_size":** write the initial window size (pixel number). The initial window size maximizes the correlation between the base and the target image. In theory, the window size should be at least twice the expected displacement, but in practice (Ayoub et al., 2017), we recommend a ratio of 64 to not waste lots of time (the larger the ratio, the longer the correlation). From the experiment, we can also choose a ratio of 128 or 256 to get a good result but it takes lots of time.
   - **(b) "step":** write the step in the direction X and Y [X, Y]. This parameter determines the step in the X and Y direction, in pixels, between two sliding windows. If the step is greater or equal to the final window size, then all measurements are independent (Ayoub et al., 2017).
   - **(c) "grid":** write "True" if you want this option or "False" if you don't want it. By writing "True" COSI-Corr will obtain a displacement map with the top-left corner coordinates to be integer multiple of the ground resolution. This is most useful when several correlations of the same area need to be overlaid or mosaiced. If unchecked the top-left corner coordinates of the displacement map will be identical to the top-left corner coordinates of the master image (Ayoub et al., 2017).

   For Frequency correlator:

   - **(d) "mask_th" (**Mask Threshold): write the mask threshold. This parameter allows the masking of the frequencies according to the amplitude of the log-cross-spectrum (Ayoub et al., 2017). A value close to unity is appropriate in most cases (Leprince et al., 2007).
   - **(e) "nb_iters"** (Iteration): write the iteration. This is the number of times per measurement the frequency mask should be adaptively re-computed. The mask contributes to reducing the noise on the measurements (Ayoub et al., 2017). 2 to 4 iterations are satisfying in most cases (Leprince et al., 2007).

   For Spatial correlator:

   - **(d) "search_range":** write the search range in the direction X and Y [X, Y]. It corresponds to the maximum distance in X and Y directions in pixels where the displacements to measure are to be searched (Ayoub et al., 2017).



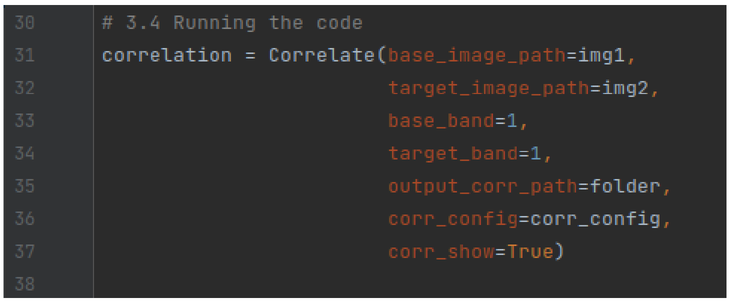
```
# 3.3 Parameters
corr_config = {"correlator_name": "frequency",
               "correlator_params": {
                   "window_size": [64, 64, 64, 64],
                   "step": [4, 4],
                   "grid": False,
                   "mask_th": 0.97,
                   "nb_iters": 4
               }
               }
```

```
# 3.3 Parameters
corr_config = {"correlator_name": "spatial",
               "correlator_params": {
                   "window_size": 4 * [64],
                   "step": [16, 16],
                   "grid": True,
                   "search_range": [10, 10]
               }
               }
```

Figure 4: Parameters for the correlation with "frequency" correlator on left and "spatial" correlator on right

3.4 Running the code

The fourth part of the code allows to run it (Figure 5). Change nothing.



```
# 3.4 Running the code
correlation = Correlate(base_image_path=img1,
                        target_image_path=img2,
                        base_band=1,
                        target_band=1,
                        output_corr_path=folder,
                        corr_config=corr_config,
                        corr_show=True)
```

Figure 5: Running the code

4. Results

After running the code the results appear in Corr_sample folder. To compare your results with the right results, you can find the right results in Corr_sample/results.

The results are composed of three documents :

1. **namebaseimage_vs_nametargetimage_correlatorname_ws_#_step_#.png:** this is the 2D correlation file in png format (Figure 6). The document is composed of two maps: East-West (left) and North-South (right) displacement. The displacement scale is in meters. For the East-West and North-South maps, the blue color matches respectively with the East and North displacement and the red color, with the West and South displacement. The axis X is the latitude and the axis Y is the longitude in the EPSG 4326 geographic coordinate system.

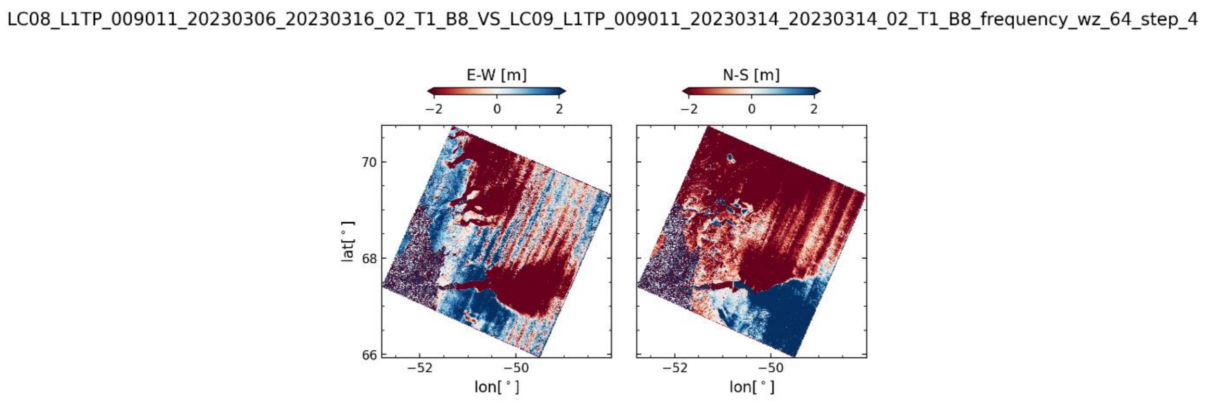

LC08_L1TP_009011_20230306_20230316_02_T1_B8_VS_LC09_L1TP_009011_20230314_20230314_02_T1_B8_frequency_wz_64_step_4
E-W [m]
−2
0
2
N-S [m]
−2
0
2
70
68
66
lat[°]
−52
−50
lon[°]
−52
−50
lon[°]

Figure 6: 2D correlation map, png format, glacier_jakobshavn example

2. **namebaseimage_vs_nametargetimage_correlatorname_ws_#_step_#.tif:** this is the 2D correlation map in the tif format. Open this file by using QGIS software. You can see the image below with the three bands: East/West (Gray), North/South and SNR (Signal Noise Ratio), (Figure 7).

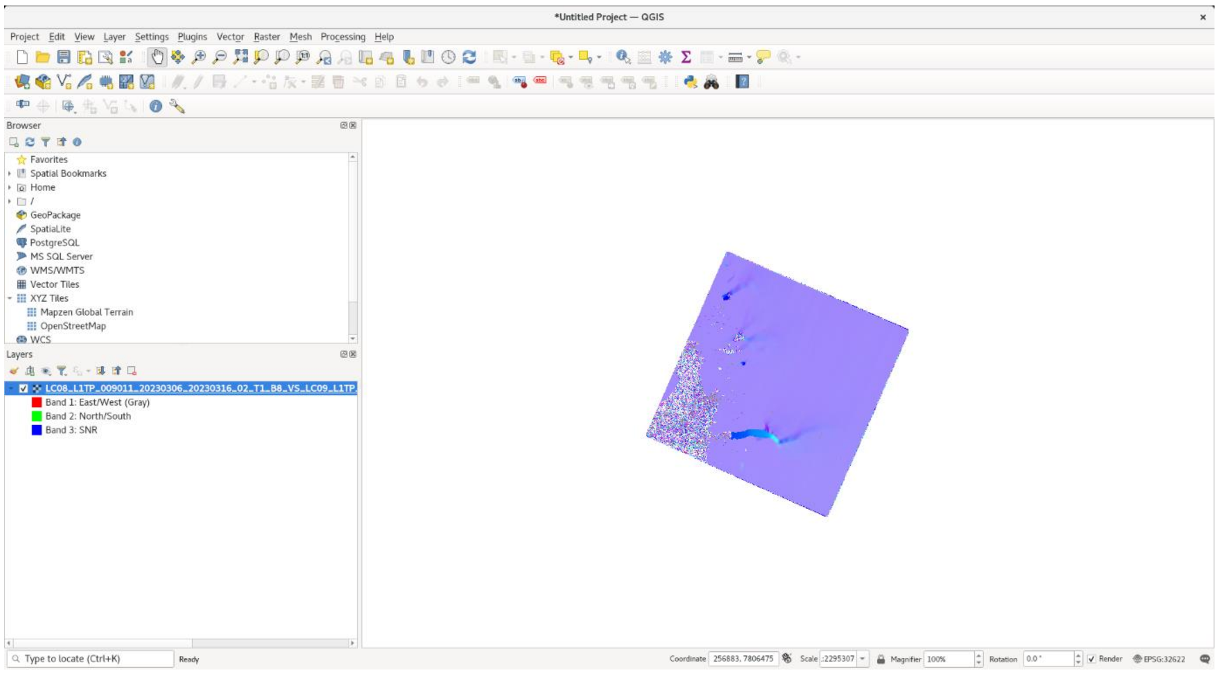


Figure 7: 2D correlation map on QGIS, glacier_jakobshavn example

Follow the instructions below to visualize the three bands:

(a) Band 1: East/West (Gray)

Right-click on the name of the image (LC08_L1TP…) then on "Properties" (Figure 8).

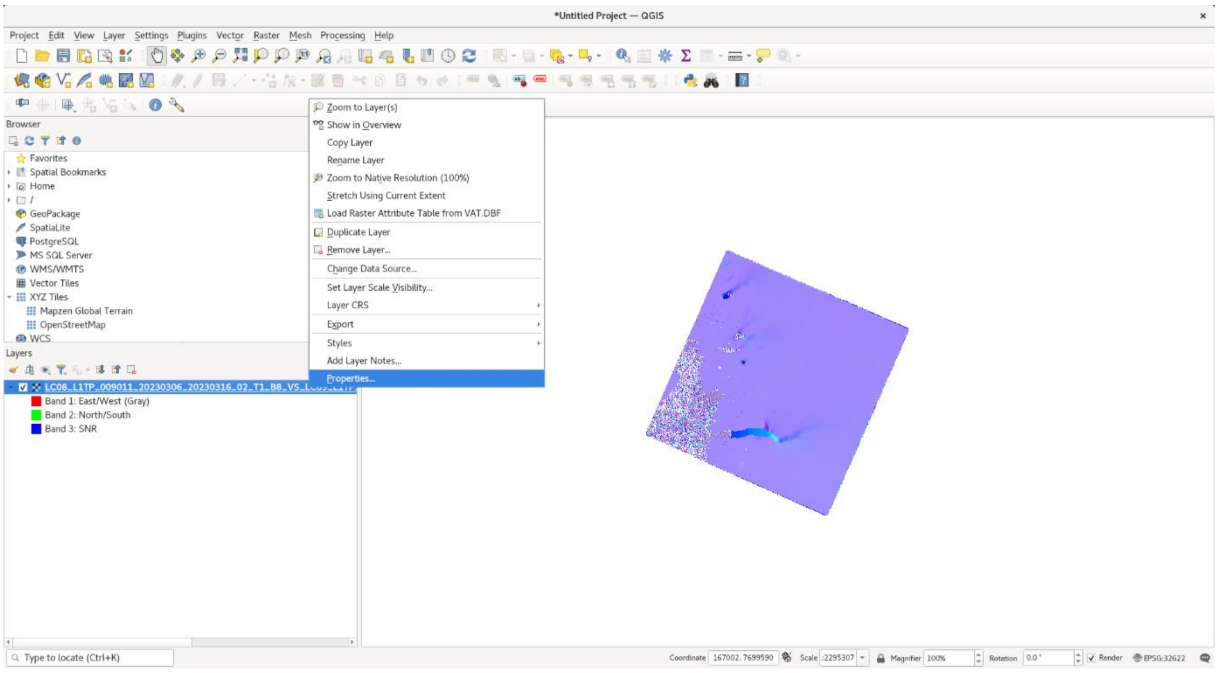
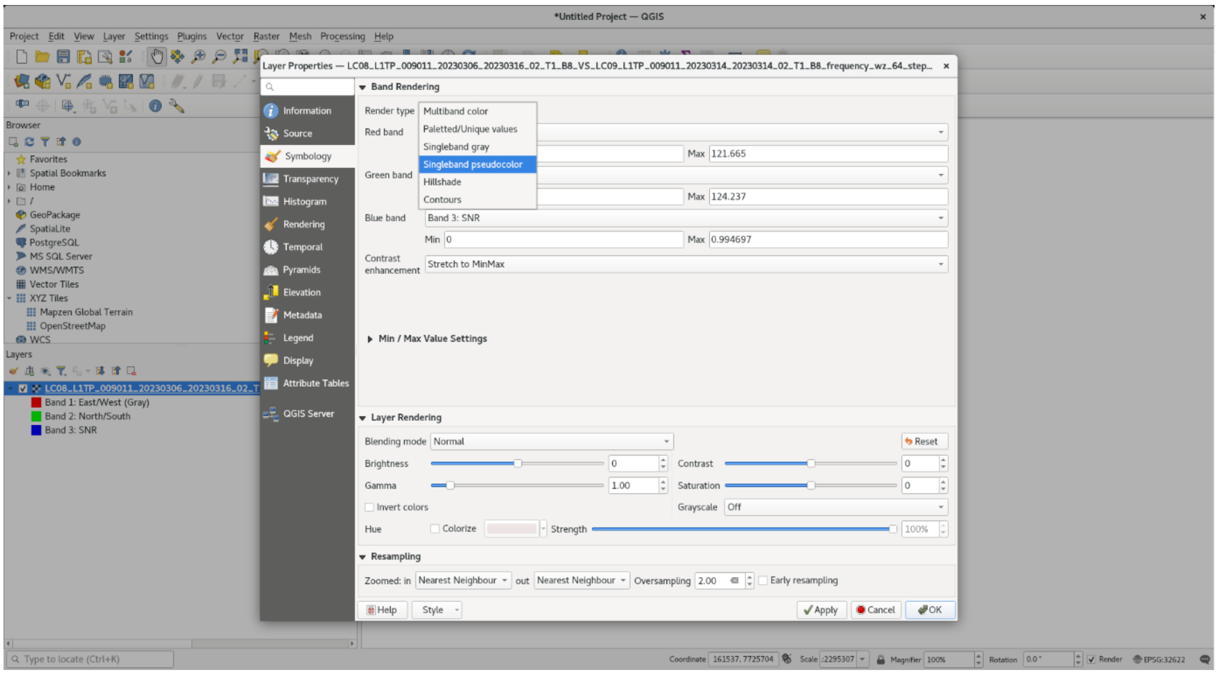


Figure 8: Path to access layer properties, glacier_jakobshavn example

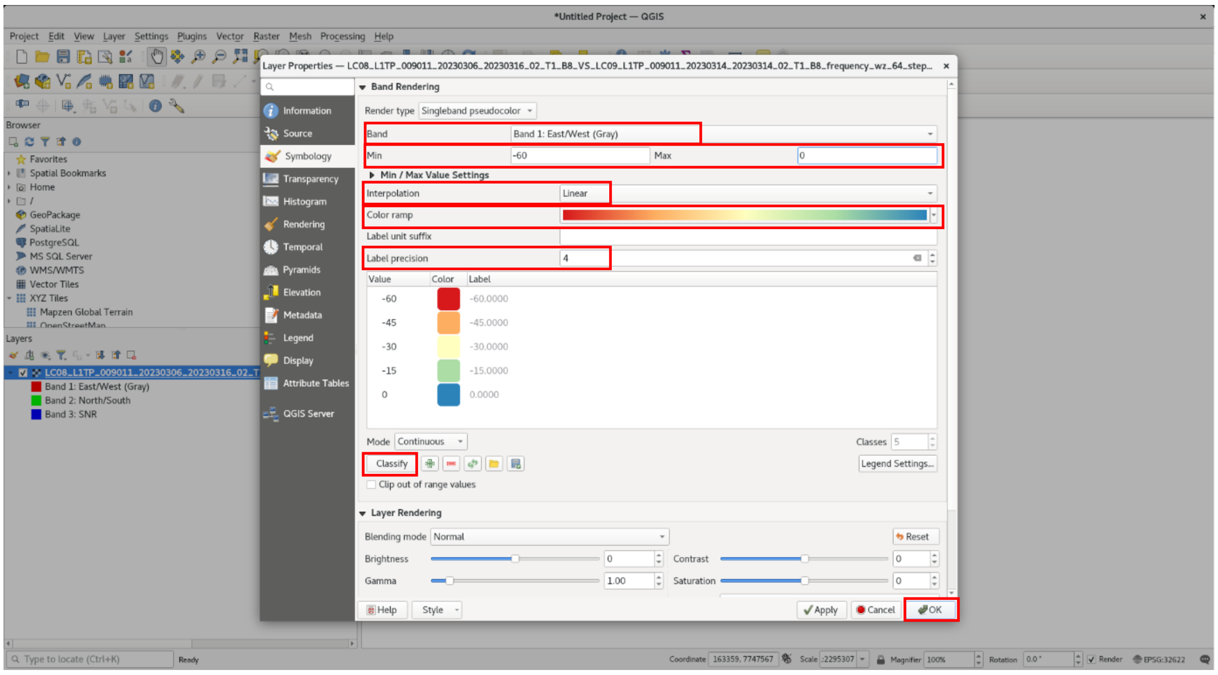
Click on "Symbology" and then select "Singleband pseudocolor", (Figure 9).



Band Rendering
Render type: Singleband pseudocolor
Red band — Max 121.665
Green band — Max 124.237
Blue band: Band 3: SNR — Min 0 — Max 0.994697
Contrast enhancement: Stretch to MinMax
Min / Max Value Settings
Layer Rendering
Blending mode: Normal
Brightness: 0 — Contrast: 0
Gamma: 1.00 — Saturation: 0
Invert colors — Grayscale: Off
Hue: Colorize — Strength: 100%
Resampling
Zoomed: in Nearest Neighbour — out Nearest Neighbour — Oversampling 2.00 — Early resampling

Figure 9: Selection of the band, glacier_jakobshavn example

Select Band 1: East/West (Gray), write the minimum (here -60) and maximum (here 0) of the scale in meters, select the interpolation, color ramp and label precision. Click on "Classify" and "Ok", (Figure 10).


Layer Properties — LC08_L1TP_009011_20230306_20230316_02_T1_B8_VS_LC09_L1TP_009011_20230314_20230314_02_T1_B8_frequency_wz_64_step...

Band Rendering

Render type: Singleband pseudocolor
Band: Band 1: East/West (Gray)
Min: -60 Max: 0
Min / Max Value Settings
Interpolation: Linear
Color ramp
Label unit suffix
Label precision: 4

Value	Color	Label
-60	🟥	-60.0000
-45	🟧	-45.0000
-30	🟨	-30.0000
-15	🟩	-15.0000
0	🟦	0.0000

Mode: Continuous Classes: 5
Classify
Clip out of range values

Layer Rendering

Blending mode: Normal
Brightness: 0 Contrast: 0
Gamma: 1.00 Saturation: 0

Figure 10: Selection of parameters of the band East/ West, glacier_jakobshavn example

We get the East/West displacement map (Figure 11).

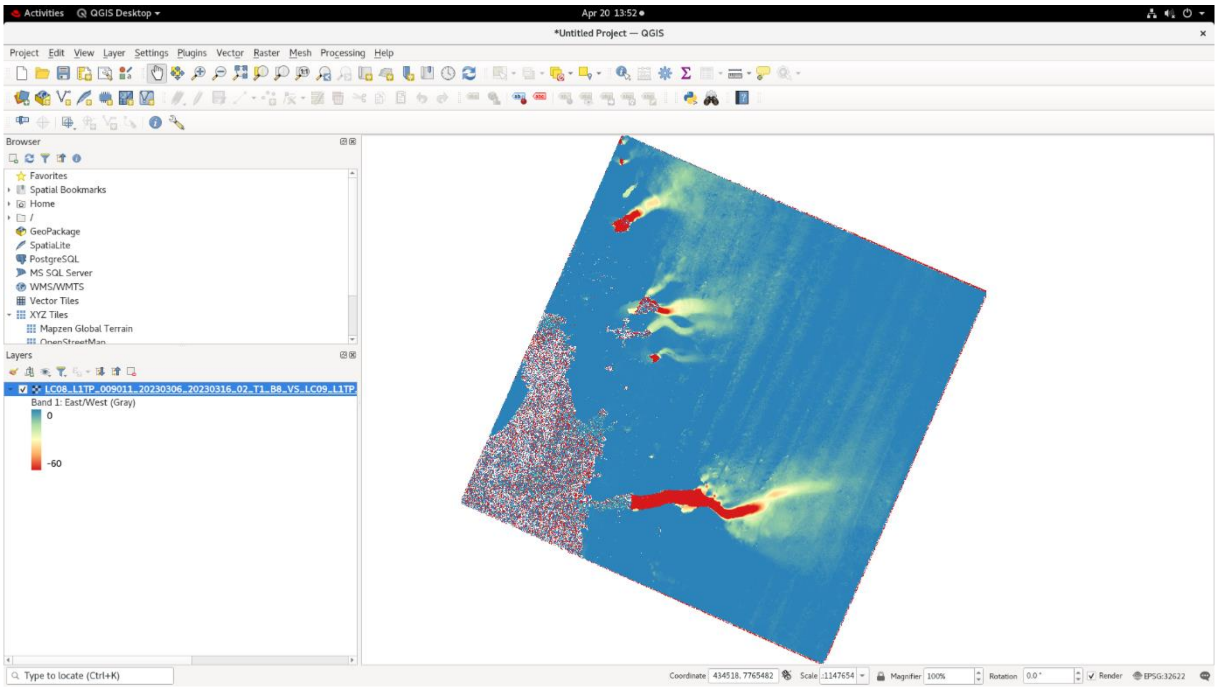
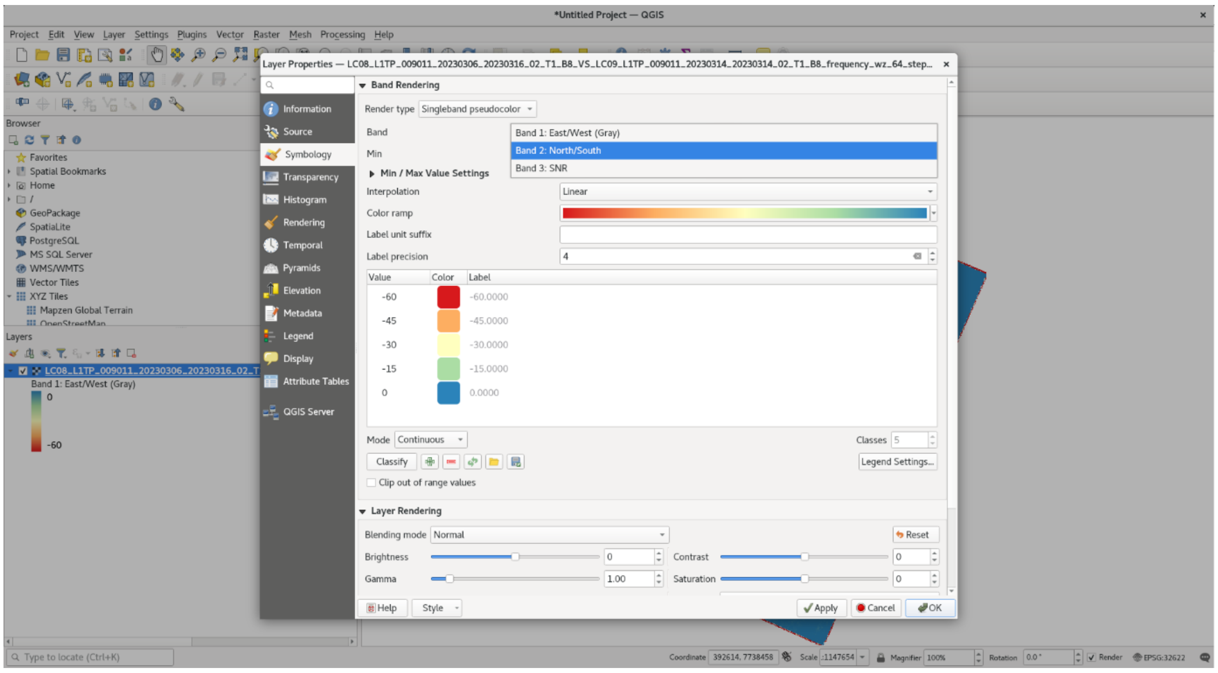


Figure 11: 2D correlation map, displacement East/West, glacier_jakobshavn example

(b) Band 2: North/South
Right-click on the name of the image (LC08_L1TP…) then on "Properties". Select Band 2: North/South and complete the parameters (Figure 12).


*Untitled Project — QGIS

Layer Properties — LC08_L1TP_009011_20230306_20230316_02_T1_B8_VS_LC09_L1TP_009011_20230314_20230314_02_T1_B8_frequency_wz_64_step...

Band Rendering

Render type: Singleband pseudocolor

Band: Band 1: East/West (Gray) / Band 2: North/South / Band 3: SNR

Interpolation: Linear

Label precision: 4

Value	Color	Label
-60	🟥	-60.0000
-45	🟧	-45.0000
-30	🟨	-30.0000
-15	🟩	-15.0000
0	🟦	0.0000

Mode: Continuous — Classes: 5

Layer Rendering

Blending mode: Normal — Brightness: 0 — Contrast: 0 — Gamma: 1.00 — Saturation: 0

Figure 12: Selection of parameters of the band North/South, glacier_jakobshavn example

We get the North/South displacement (Figure 13).

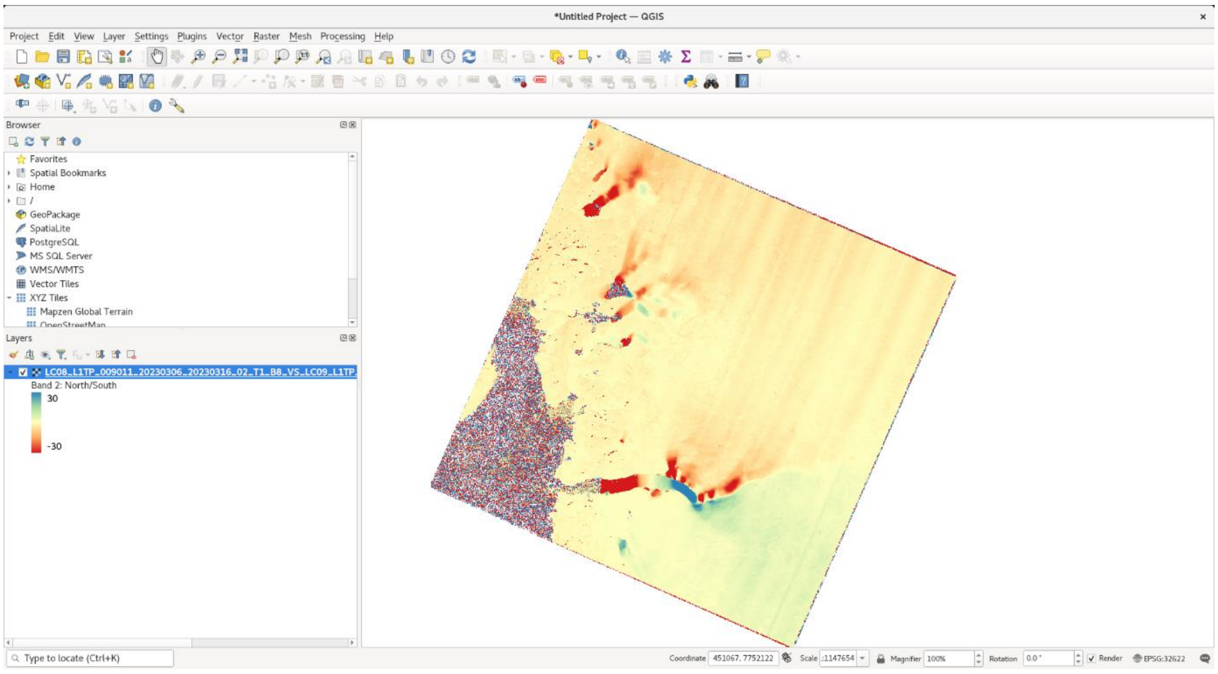
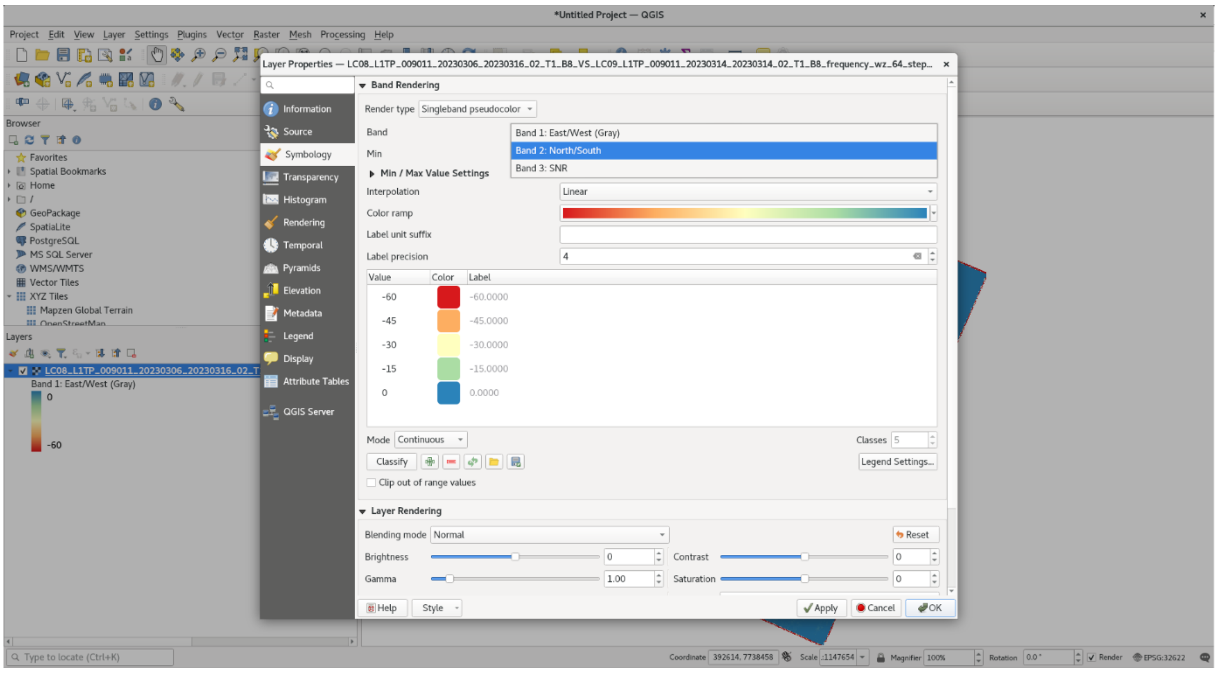

*Untitled Project — QGIS

Figure 13: 2D correlation map, displacement North/South, glacier_jakobshavn example

(c) Band 3: SNR (Signal Noise Ratio)
The SNR value is included between 0 and 1. A SNR inferior at 1 indicates more noise than signal in the area and so a bad correlation. Studying the SNR map help interpret Band 1 and 2.

Right-click on the name of the image (LC08_L1TP…) then on "Properties". Select Band 3: SNR and complete the parameters with a minimum equal to 0 and a maximum equal to 1 (Figure 14).


*Untitled Project — QGIS

Layer Properties — LC08_L1TP_009011_20230306_20230316_02_T1_B8_VS_LC09_L1TP_009011_20230314_20230314_02_T1_B8_frequency_wz_64_step...

Band Rendering

Render type: Singleband pseudocolor

Band: Band 1: East/West (Gray) / Band 2: North/South / Band 3: SNR

Interpolation: Linear

Label precision: 4

Value	Color	Label
-60	🟥	-60.0000
-45	🟧	-45.0000
-30	🟨	-30.0000
-15	🟩	-15.0000
0	🟦	0.0000

Mode: Continuous — Classes: 5

Layer Rendering

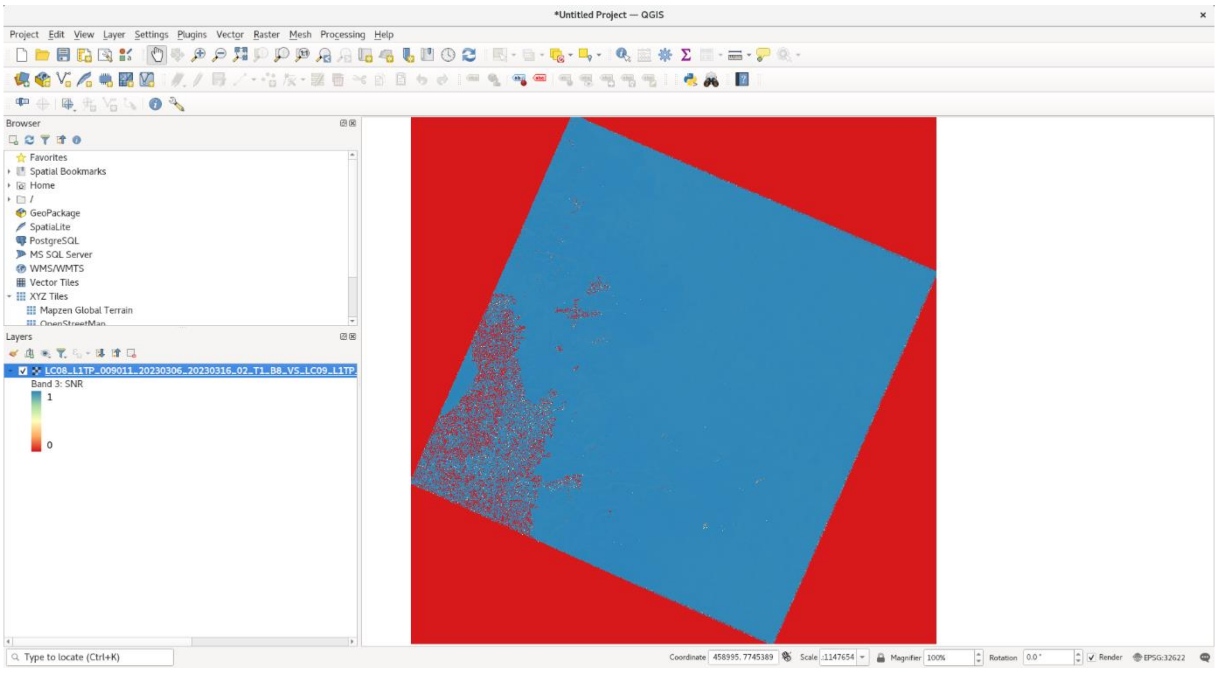
Blending mode: Normal — Brightness: 0 — Contrast: 0 — Gamma: 1.00 — Saturation: 0

Layers: LC08_L1TP_009011_20230306_20230316_02_T... — Band 1: East/West (Gray) — 0 / -60

Coordinate 392614, 7738458 — Scale 1:1147654 — Magnifier 100% — Rotation 0.0° — EPSG:32622

Figure 14 : Selection of parameters of the band SNR, glacier_jakobshavn example

You get the SNR map (Figure 15).


*Untitled Project — QGIS

Layers: LC08_L1TP_009011_20230306_20230316_02_T1_B8_VS_LC09_L1TP — Band 3: SNR — 1 / 0

Coordinate 458995, 7745389 — Scale 1:1147654 — Magnifier 100% — Rotation 0.0° — EPSG:32622

Figure 15 : 2D correlation map, band SNR, glacier_jakobshavn example

3. **namebaseimage_vs_nametargetimage_correlatorname_ws_#_step_#_projection.vrt:** document containing all the information about the coding process.

5. GUI correlation

To get the 2D correlation map, it is possible to use the GUI (Graphical User Interface): Geospatial-COSICorr3D/geoCosiCorr3D/geoCosiCorr3D_GUI/correlate_GUI. The GUI uses the same code as in this example, but it is a different way to enter the input parameters. In the GUI, there is one more option "Batch Correlator" which allows to perform multiple correlations between multiple images. To get more information about the GUI, refer to GUI_DOC.md in Geospatial-COSICorr3D/Doc.

6. References

Ayoub, F., Leprince, S., Avouac, J.-P., 2017. User's Guide to COSI-CORR Co-registration of Optically Sensed Images and Correlation. California Institute of Technology, Pasadena, CA.

Leprince, S., Barbot, S., Ayoub, F., Avouac, J.-P., 2007. Automatic and Precise Ortho-rectification, Coregistration, and Subpixel Correlation of Satellite Images, Application to Ground Deformation Measurements.