

# User's Guide to COSI-CORR Python code

## Co-registration of Optically Sensed Images and Correlation

### Orthorectification Example

Marie Cadoux, Saif Aati, Jean-Philippe Avouac

California Institute Of Technology  
1200 East California Blvd, Pasadena, CA 91125, USA

August, 2023

#### **Abstract**

This document is a user's guide to orthorectify three SPOT 2 images at Izmit. All the input parameters and results are present to help the user to understand the code. Four examples are treated by changing the input parameters with the three same SPOT 2 images. This code is also adapted to orthorectify images that come from other sensors: SPOT 1 to SPOT 7, WorldView 1 to WorldView 4, GoeEye, QuickBird, DigitalGlobe, SkySat, PlanetScope Dove-C, PlanetScope Dove-R and PlanetScope Super Dove sensors.

## Contents

1.	Introduction .....	3
2.	Structure of the example.....	3
3.	Code operation .....	3
3.1	Import classes and functions .....	4
3.2	Paths input parameters.....	4
3.3	Documents with the input parameters .....	4
3.4	Running the code.....	10
4.	Results .....	11
5.	References .....	14

# 1. Introduction

Orthorectification\_example is a file containing one example of orthorectification of three SPOT 2 images at Izmit. As a reminder, orthorectification allows to correct of the stereoscopic effects (geometric distortions) on an image called “raw image” by resampling it in a UTM grid.

All the input parameters and results are present in this example. By changing the input parameters, four different results are available: output, output\_correction\_model\_file, output\_DEM and output\_GCPs. The goal of this folder is to help the user with the first uses of the code.

## 2. Structure of the example

orthorectification\_example contains different files:

- input\_izmit.json, input\_izmit\_correction\_model\_file.json, input\_izmit\_DEM.json and input\_izmit\_GCPs.json: documents containing the input parameters,
- ortho\_example.py: code,
- REF\_DATA: folder containing the reference image called rOrtho\_1999-07-Spot4.tif and the DEM called SRTM\_DEM.tif (rOrtho\_1999-07-Spot4\_fp.geojson, Rupture\_ROI.geojson, SRTM\_DEM\_32635.vrt and SRTM\_DEM\_32636.vrt are created after running the code),
- SP2\_DS: folder containing the three SPOT 2 images called IMAGERY.TIF with their metadata called METADATA.DIM (METADA.pkl is created after running the code),
- Results: folder containing different results (output, output\_correction\_model\_file, output\_DEM, output\_GCPs) of orthorectification according to the input parameters.

## 3. Code operation

ortho\_example.py is the code allowing to orthorectify images (Figure 1).

```
6 # Import classes and functions
7 from geoCosiCorr3D.geoCosiCorr3D_scripts.script_orthorectification import *
8
9
10 if __name__ == '__main__':
11
12     # Paths input parameters
13     inputs_file = '/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/input_izmit_correction_model_file.json'
14     config_file = "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D/geoCore/geoCosiCorrBaseCfg/geo_ortho_config.yaml"
15
16     # Running the code
17     main_geoOrtho(input_file=inputs_file, config_file=config_file)
```

Figure 1: Ortho\_example.py

The code is organized into different parts:

- Import classes and functions,
- Paths input parameters,
- Documents with the input 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 that allow running `ortho_example.py` (Figure 2).

```
6 # Import classes and functions
7 from geoCosiCorr3D.geoCosiCorr3D_scripts.script_orthorectification import *
```

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

### 3.2 Paths input parameters

The second part of the code consists of writing the different paths of the documents containing the input parameters and the configuration of the code (Figure 3).

1. **inputs\_file = 'home/...':** write the path to assign `inputs_file` to the document containing the input parameters: `input_izmit.json`, `input_izmit_GCPS.json`, `input_izmit_DEM.json` or `input_izmit_correction_model_file.json` in this example.
2. **config\_file = '.../geo\_ortho\_config.yaml':** write the path to assign `geo_ortho_config.yaml` to `config_file`.

```
12 # Paths input parameters
13 inputs_file = '/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/input_izmit_correction_model_file.json'
14 config_file = '/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D/geoCore/geoCosiCorrBaseCfg/geo_ortho_config.yaml'
```

Figure 3: Paths of the files that contain the input parameters for the orthorectification

### 3.3 Documents with the input parameters

After writing the paths to access the different documents containing the input parameters and the configuration of the code, it is necessary to complete it.

For the input parameters, four documents, with different input parameters but the same code are available in this example: `input_izmit.json`, `input_izmit_correction_model_file.json`, `input_izmit_DEM.json`, and `input_izmit_GCPS`.

1. **input\_izmit\_correction\_model\_file.json:** this file allows getting orthorectified image(s) with a correction model file (Figure 4). As a reminder, a correction model file optimizes the information about the imaging system of the raw image to reduce the error location of each GCPs between the raw image(s) and the reference image, by using the method RSM or RFM refinement.
  - (a) **“raw\_image\_path”:** “...”: write the path of the raw image(s). You can write several paths. In this example, we have three SPOT 2 raw images so three paths: one path for each image.
  - (b) **“metadata\_file\_path”:** “...”: write the path of the document containing information about the imaging system(s) of the raw image. In this example, we have three SPOT 2 raw images so three paths of the document containing information about the imaging systems: one path for each image.

- (c) **“dem\_path”**: **“...”**: write the path of the DEM. To get orthorectified image(s) with a correction model file, it is necessary to write the path of the DEM. Thanks to a DEM the code can assign the elevation coordinate for each pixel in the raw image. Either it is not mandatory.
- (d) **“ref\_ortho\_path”**: **“...”**: write the path of the reference image (image already orthorectified). To get orthorectified image(s) with a correction model file, it is necessary to write the path of the reference image. Thanks to a reference image the code will select GCPs between the raw image(s) and the reference image to orthorectify the raw image in the same condition as the reference image. Either it is not mandatory.
- (e) **“log\_name”**: **“RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW”**: affects the name of the document, in the log format, that contains the coding process. This document is located in the path written in the **“workspace\_folder”** (h). We can change the name of this document.
- (f) **“metadata\_type”**: **RSM**: write the refinement method to orthorectify. There are two options: RSM (Rigorous Sensor Modeling) and RFM (Rational Function Model). The choice of the method depends on the sensor. In this example, this is SPOT 2 sensor and the refinement method adapted for this sensor is RSM.
- (g) **“sensor”**: **“Spot2”**: write the name of the sensor that captured the raw image(s). In this example, the three raw images come from SPOT 2 sensor. The list of sensors that we can use is available in constants.py code (Geospatial-COSICorr3D/GeoCosiCorr3D/geoCore/constants.py).
- (h) **“workspace\_folder”**:  
**“.../geoCosiCorr3D\_examples/orthorectification\_example/results/output\_correction\_model\_file”**: write the path of the folder where we want to stock the results. In this example, the folder containing the results is called **“output\_correction\_model\_file”**.
- (i) **“ortho\_gsd”**: **10**: write the GSD (Ground Sampling Distance). It is possible to choose a GSD inferior to the resolution of the raw images but the orthorectified images got, will have more artifacts. In this example, Spot 2 images have a resolution of 10m so we can choose at least the GSD of 10 to get orthorectified images without adding artifacts.
- (j) **“correction\_model\_file”**: **“”**: write the path of the correction model file (txt format) if we have it. A correct model file allows to get better orthorectification by optimizing the information about the imaging system(s). In this example, we don’t have this file.
- (k) **“optimize”**: **true**: creates a correction model file, by writing **“true”**, if we don’t have it in (j). The code will select GCPs between the raw image(s) and the reference image to orthorectify the raw image(s).

After, the code optimizes the information about the imaging system(s) of raw image(s). Thanks to the optimization COSI-Corr reduces the error location of each GCPs to get a better orthorectification. By default, the code does three loops to optimize. It is possible to change the number of loops in `geo_ortho_config.yaml`. To write **“true”** and get a correction model file, we must have a DEM and a reference image. Either we must write **“false”**.

```

1  {
2      "raw_img_path": [
3          "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-006_S2_103-268-0_1998-06-10-09-01-06_HRV-1_P_DT_TT/IMAGERY.TIF",
4          "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-03-03-09-05-04_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF",
5          "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF"
6      ],
7      1,
8
9
10     "metadata_file_path": [
11         "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-006_S2_103-268-0_1998-06-10-09-01-06_HRV-1_P_DT_TT/METADATA.DIM",
12         "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-03-03-09-05-04_HRV-1_P_DT_TT/SCENE01/METADATA.DIM",
13         "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/METADATA.DIM"
14     ],
15     1,
16
17
18     "dem_path": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/REF_DATA/SRM_DEM.tif",
19     "ref_ortho_path": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/REF_DATA/rOrtho_1999-07-Spot4.tif",
20     "log_name": "RSM_ORTHO_SYSTEMATIC_SP2_WORKFLOW",
21     "metadata_type": "RSM",
22     "sensor": "Spot2",
23     "workspace_folder": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/results/output_correction_model_file",
24     "ortho_gsd": 10,
25     "correction_model_file": "",
26     "optimize": true
27 }

```

Figure 4: *input\_izmit\_correction\_model\_file.json*, a file that contains the inputs parameters to get orthorectified image(s) with a correction model file

2. **input\_izmit\_GCPS.json:** this file allows getting folder(s) that contain the tie points and GCPs (without the elevation coordinate) between the raw image(s) and a reference image (Figure 5). Tie points are common points between raw image(s) and a reference image but the coordinates are just the pixel location. GCPs are common points between raw image(s) and a reference image with the coordinates in a geographic system.
  - (a) **"raw\_image\_path": "...":** write the path of the raw image(s).
  - (b) **"metadata\_file\_path": "...":** write the path of the document(s) containing information about the imaging system(s) of the raw image(s).
  - (c) **"dem\_path": "":** to get the folder(s) containing the tie points and the GCPs (without the elevation coordinate), don't use a DEM and write nothing between the quotation marks.
  - (d) **"ref\_ortho\_path": "...":** write the path of the reference image. To get the folder(s) containing the tie points and the GCPs (without the altitude coordinate), it is necessary to write the path of the reference image. Thanks to a reference image the code will select GCPs between the raw image(s) and the reference image. Either it is not mandatory.
  - (e) **"log\_name": "RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW":** affects the name of the document, in the log format containing the coding process. This document is located in the path written in the "workspace\_folder" (h).
  - (f) **"metadata\_type": RSM:** write the refinement method to orthorectify. There are two options: RSM (Rigorous Sensor Modeling) and RFM (Rational Function Model).
  - (g) **"sensor": "Spot2":** write the name of the sensor that captured the raw images.
  - (h) **"workspace\_folder":**  
**"../geoCosiCorr3D\_examples/orthorectification\_example/results/output\_GCPs":**  
write the path of the folder where we want to stock the results. In this example, the folder containing the results is called "output\_GCPs".
  - (i) **"ortho\_gsd": 10:** write the GSD (Ground Sampling Distance).
  - (j) **"correction\_model\_file": "":** write the path of the correction model file (txt format) if we have it. In this example, we don't have this file.
  - (k) **"optimize": true:** by writing "true", the code will create file(s) that contain the tie points and GCPs (without the altitude coordinate) between the raw image(s) and a reference image. To write "true" and get the file(s) containing only the tie points and GCPs (without the elevation coordinate), the code must have a reference image. Either you can write "false".

```

1 {
2   "raw_img_path": [
3     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_DS/003-006_S2_103-268-0_1998-06-18-09-01-06_HRV-1_P_DT_TT/IMAGERY.TIF",
4     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_DS/003-008_S2_103-268-0_1998-03-03-09-05-04_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF",
5     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_DS/003-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF"
6   ],
7   1,
8
9   "metadata_file_path": [
10     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_DS/003-006_S2_103-268-0_1998-06-18-09-01-06_HRV-1_P_DT_TT/METADATA.DIN",
11     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_DS/003-008_S2_103-268-0_1998-03-03-09-05-04_HRV-1_P_DT_TT/SCENE01/METADATA.DIN",
12     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_DS/003-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/METADATA.DIN"
13   ],
14   1,
15
16   "dem_path": "",
17   "ref_ortho_path": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/REF_DATA/r0ortho_1999-07-Spot4.tif",
18   "log_name": "RSM_ORTHO_SYSTEMATIC_SP2_WORKFLOW",
19   "metadata_type": "RSM",
20   "sensor": "Spot2",
21   "workspace_folder": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/results/output_GCPs",
22   "ortho_gsd": 10,
23   "correction_model_file": "",
24   "optimize": true
25 }

```

Figure 5: *input\_izmit\_GCPs.json*, a file that contains the inputs parameters to get the tie points and GCPs between a reference image and raw image(s)

3. **input\_izmit\_DEM.json**: this file allows getting orthorectified image(s) by adding elevation coordinates without a correction model file and a reference image (Figure 6).
  - (a) **“raw\_image\_path”**: “...”: write the path of the raw image(s).
  - (b) **“metadata\_file\_path”**: “...”: write the path of the document(s) containing information about the imaging system(s) of the raw image(s).
  - (c) **“dem\_path”**: “”: write the path of the DEM. To get orthorectified image(s) by adding the elevation coordinates, it is necessary to write the path of the DEM. Thanks to a DEM the code can assign the elevation coordinate for each pixel in the raw image. Either it is not mandatory.
  - (d) **“ref\_ortho\_path”**: “”: to orthorectify without a reference image, write nothing between the quotation marks.
  - (e) **“log\_name”**: **“RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW”**: affects the name of the document, in the log format, containing the coding process. This document is located in the path written in the “workspace\_folder” (h).
  - (f) **“metadata\_type”**: **RSM**: write the refinement method to orthorectify. There are two options: RSM (Rigorous Sensor Modeling) and RFM (Rational Function Model).
  - (g) **“sensor”**: **“Spot2”**: write the name of the sensor that captured the raw images.
  - (h) **“workspace\_folder”**:  
**“.../geoCosiCorr3D\_examples/orthorectification\_example/results/output\_DEM”**: write the path of the folder where we want to stock the results. In this example, the folder containing the results is called “output”.
  - (i) **“ortho\_gsd”**: **10**: write the GSD (Ground Sampling Distance).
  - (j) **“correction\_model\_file”**: “”: because we don’t have a reference image, we can’t write a path for the correction model file to orthorectify raw image(s).
  - (k) **“optimize”**: **false**: because we don’t have a reference image, we can’t create a correction model file to orthorectify raw image(s). If we write “true”, the code will not work.

```

1 4
2 "raw_img_path": [
3     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-006_S2_103-268-0_1998-06-10-09-01-06_HRV-1_P_DT_TT/IMAGERY.TIF",
4     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-03-01-09-05-04_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF",
5     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF"
6 ],
7
8
9
10 "metadata_file_path": [
11     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-006_S2_103-268-0_1998-06-10-09-01-06_HRV-1_P_DT_TT/METADATA.DIM",
12     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-03-01-09-05-04_HRV-1_P_DT_TT/SCENE01/METADATA.DIM",
13     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/003-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/METADATA.DIM"
14 ],
15
16
17
18
19 "dem_path": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/REF_DATA/SRTM_DEM.tif",
20 "ref_ortho_path": "",
21 "log_name": "RSM_ORTHO_SYSTEMATIC_SP2_WORKFLOW",
22 "metadata_type": "RSM",
23 "sensor": "Spot2",
24 "workspace_folder": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/results/output_DEM",
25 "ortho_gsd": 10,
26 "correction_model_file": "",
27 "optimize": false
28 }

```

Figure 6: *input\_izmit\_DEM.json*, a file that contains the inputs parameters to get orthorectified image(s) by adding the elevation coordinates and without a reference image and a correction model file

4. **input\_izmit.json**: this file allows getting orthorectified images without a correction model file, reference image and without adding elevation coordinates (Figure 7).
  - (a) **“raw\_image\_path”**: “...”: write the path of the raw image(s).
  - (b) **“metadata\_file\_path”**: “...”: write the path of the document(s) containing information about the imaging system(s) of the raw image(s).
  - (c) **“dem\_path”**: “”: to orthorectify without adding the elevation coordinates, don’t use a DEM and write nothing between the quotation marks.
  - (d) **“ref\_ortho\_path”**: “”: to orthorectify without a reference image, write nothing between the quotation marks.
  - (e) **“log\_name”**: **“RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW”**: affects the name of the document, in the log format, containing the coding process. This document is located in the path written in the “workspace\_folder” (h).
  - (f) **“metadata\_type”**: **RSM**: write the refinement method to orthorectify. There are two options: RSM (Rigorous Sensor Modeling) and RFM (Rational Function Model).
  - (g) **“sensor”**: **“Spot2”**: write the name of the sensor that captured the raw images.
  - (h) **“workspace\_folder”**: **“.../geoCosiCorr3D\_examples/orthorectification\_example/results/output”**: write the path of the folder where we want to stock the results. In this example, the folder containing the results is called “output”.
  - (i) **“ortho\_gsd”**: **10**: write the GSD (Ground Sampling Distance).
  - (j) **“correction\_model\_file”**: “”: because we don’t have DEM and a reference image, we can’t write a path for the correction model file to orthorectify raw image(s).
  - (k) **“optimize”**: **false**: because we don’t have DEM and a reference image, we can’t create a correction model file to orthorectify raw image(s). If we write “true”, the code will not work.



```

1 {
2   "raw_img_path": [
3     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/803-006_S2_103-268-0_1998-06-10-09-01-06_HRV-1_P_DT_TT/IMAGERY.TIF",
4     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/803-008_S2_103-268-0_1998-03-03-09-05-04_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF",
5     "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/803-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/IMAGERY.TIF"
6   ],
7   1,
8
9
10  "metadata_file_path": [
11    "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/803-006_S2_103-268-0_1998-06-10-09-01-06_HRV-1_P_DT_TT/METADATA.DIM",
12    "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/803-008_S2_103-268-0_1998-03-03-09-05-04_HRV-1_P_DT_TT/SCENE01/METADATA.DIM",
13    "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/SP2_05/803-008_S2_103-268-0_1998-04-03-09-08-28_HRV-1_P_DT_TT/SCENE01/METADATA.DIM"
14  ],
15  1,
16
17
18  "dem_path": "",
19  "ref_ortho_path": "",
20  "log_name": "RSM_ORTHO_SYSTEMATIC_SP2_WORKFLOW",
21  "metadata_type": "RSM",
22  "sensor": "Spot2",
23  "workspace_folder": "/home/cosicorr/PycharmProjects/Geospatial-COSICorr3D/geoCosiCorr3D_examples/orthorectification_example/results/output",
24  "ortho_gsd": 10,
25  "correction_model_file": "",
26  "optimize": false
27 }

```

Figure 7: *input\_izmit.json*, a file that contains the inputs parameters to get orthorectified image(s) without a DEM, a reference image and a correction model file

geo\_ortho\_config.yaml is also necessary to run orthorectification\_example. Follow the path to find this file:

‘Geospatial-COSICorr3D/GeoCosiCorr3D/geoCore/geoCosiCorrBAseCfg/geo\_ortho\_config.yaml’.

5. **geo\_ortho\_config.yaml:** file presenting all the functions with the different input parameters used to orthorectify raw image(s), (Figure 8). We can modify the parameters of the different functions in this file.

```
1 feature_points_params:
2   method: 'ASIFT'
3   kernel:
4   scale_factor: 0.5
5   matching_filter: False
6   max_pts: 50
7
8 opt_corr_config:
9   correlator_name: "frequency"
10  correlator_params:
11    window_size: [ 64,64,64,64 ]
12    step: [ 8, 8 ]
13    grid: True
14    mask_th: 0.95
15    nb_iters: 4
16
17 sat_model_params:
18   sat_model:
19   metadata:
20   sensor:
21
22 opt_params:
23   nb_loops: 3
24   snr_th: 0.9
25   mean_error_th: 0.2
26   resampling_method: 'sinc'
27
28 ortho_params:
29   method:
30     method_type: "RSM"
31     metadata:
32     sensor:
33     corr_model:
34
35   GSD: 10
36   resampling_method: "sinc"
```

Figure 8 : geo\_ortho\_config.yaml, file presenting the input parameters of the different functions used to orthorectify

### 3.4 Running the code

The third part of the code allows us to run it by using the function main\_geoOrtho (Figure 9). Change nothing.

```
16 # Running the code
17 main_geoOrtho(input_file=inputs_file, config_file=config_file)
18
```

Figure 9: Function used to orthorectify image(s)

## 4. Results

After running the code the results appear in the path we have written in “workspace\_folder” (h) in the document that contains the input parameters (.json format). To compare your results with the right results, you can find the right results in orthorectification\_example/results.

The results are composed of four documents :

1. **output:** folder containing the results from input parameters written in input\_izmit.json. In output we can find:
  - (a) **635ef8da-eabd-11ed-b16b-4cd98f389017\_IMAGERY:** this folder contains:
    - IMAGERY\_ORTHO.tif: orthorectified image that comes from the SPOT 2 image 003-008\_S2\_103-268-0\_1998-03-03-09-05-04\_HRV-1\_P\_DT\_TT,
    - IMAGERY\_TRX.tif: this is the transformation matrix used for the orthorectification. This matrix contains Band-1 and Band-2 which correspond respectively to the 2D matrices of  $X(x_{pix}, y_{pix})$  and  $Y(x_{pix}, y_{pix})$  coordinates of the pixel  $p(x_{pix}, y_{pix})$  in the raw image to be projected (Aati et al., 2022).
  - (b) **4160d6b8-eabd-11ed-b16b-4cd98f389017\_IMAGERY:** this folder contains the same kind of results as (a) but that comes from the SPOT 2 image 003-006\_S2\_103-268-0\_1998-06-10-09-01-06\_HRV-1\_P\_DT\_TT.
  - (c) **8841bbba-eabd-11ed-b16b-4cd98f389017\_IMAGERY:** this folder presents the same kind of results as (a) but that comes from the SPOT 2 image 003-008\_S2\_103-268-0\_1998-04-03-09-08-28\_HRV-1\_P\_DT\_TT.
  - (d) **geo\_ortho\_config.yaml:** this file presents all the functions with the different input parameters used to orthorectify image(s). This is the same file mentioned in 3.3.
  - (e) **input\_izmit.json:** this file contains the input parameters to get orthorectified images without a correction model file. This is the same file mentioned in 3.3.
  - (f) **RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW\_date\_hour.log:** this file presents the coding process. We have written the name of this file in input\_izmit.json, line “log\_name”, 3.3.
2. **output\_correction\_model\_file:** folder containing the results from input parameters written in input\_izmit\_correction\_model\_file.json. In output\_correction\_model\_file we can find:
  - (a) **571bcc44-eaca-11ed-a937-4cd98f389017\_IMAGERY:** this folder contains:
    - CoregistrationError.png: graph that shows the coregistration error with the number of iterations (number of the loops) for the axis X and the error (location error of GCPs between the raw image and the reference image, it is measured in the number of pixels) for the axis Y. The red curve is the average of the dispersion of the error distribution of each GCP from the origin of the error\_distribution\_loop\_#.png graph. The green curve is the distance of the red point from the origin of the error\_distribution\_loop\_#.png graph. The best is to have the two curves close to 0 at the last loop.  
If this graph doesn't appear, write “True” in geoCosiCorr3D/geo\_CosiCorr3D\_scripts/geoCosiCorr3D\_end\_2\_end\_pipeline.py L257 and L258 (Figure 10),
    - error\_distribution\_loop\_#.png: graphs that show the error distribution loop by loop. The blue points are the error distribution of each GCP. The red point is the average of the error distribution of all GCP.  
If these graphs don't appear, write “True” in geoCosiCorr3D/geo\_CosiCorr3D\_scripts/geoCosiCorr3D\_end\_2\_end\_pipeline.py L257 and L258 (Figure 10),

```

247 Path(rsm_refinement_dir).mkdir(parents=True, exist_ok=True)
248 opt = cGCP0optimization(gcp_file_path=gcp_file,
249                          raw_img_path=data0f.loc[validIndex, "ImgPath"],
250                          ref_ortho_path=self.ref_ortho,
251                          sat_model_params=sat_model_params,
252                          dem_path=self.dem_file,
253                          opt_params=self.config['opt_params'],
254                          opt_gcp_file_path=os.path.join(rsm_refinement_dir,
255                                                         Path(gcp_file).stem + "_opt.pts"),
256                          corr_config=self.config['opt_corr_config'],
257                          debug=True,
258                          svg_patches=True)

```

Figure 10: Creation of the graphs

- IMAGERY\_ORTHO.tif: orthorectified image that comes from the SPOT 2 image 003-008\_S2\_103-268-0\_1998-03-03-09-05-04\_HRV-1\_P\_DT\_TT,
  - IMAGERY\_TRX.tif: this is the transformation matrice used for the orthorectification. This matrice contains Band-1 and Band-2 which correspond respectively to the 2D matrices of  $X(x_{pix}, y_{pix})$  and  $Y(x_{pix}, y_{pix})$  coordinates of the pixel  $p(x_{pix}, y_{pix})$  in the raw image to be projected (Aati et al., 2022),
  - rOrtho\_1999-07-Spot4\_VS\_IMAGERY.pts: file that contains the list of the coordinates (X, Y) of tie points (common points) between the reference image (called “Base Image” in the file) and the raw image (called “Wrap Image” in the file). Each line corresponds to a tie point. The first column is the list of the X coordinates of the Base Image, the second column is the list of the Y coordinates of the Base Image, the third column is the list of the X coordinates of the Wrap Image and the fourth column is the list of the Y coordinates of the Wrap image. The Base Image coordinates and the Wrap Image coordinates are the position of the pixel with as origin in the upper left part of the image. COSI-Corr uses MicMac software to find automatically the tie points between the reference image and the raw image,
  - rOrtho\_1999-07-Spot4\_VS\_IMAGERY\_GCP.csv: file that contains information about each GCP. After getting the list of tie points the code will transform the tie points in GCPs. The difference between the tie points and GCPs is the coordinates. The GCPs have (X, Y, Z) projected in a geographic system. GCPs are used to orthorectify raw image(s) from a reference image and a DEM,
  - rOrtho\_1999-07-Spot4\_VS\_IMAGERY\_GCP.csv.png: graph which locates each GCP in relation to the DEM and the reference image. The elements are projected in EPSG: 4326 (WGS),
  - rOrtho\_1999-07-Spot4\_VS\_IMAGERY\_GCP\_opt.opt\_report.csv: file containing the new coordinates corrected of each GCP, by loop with the error distribution. The coordinates are corrected thanks to the RSM or RFM refinement,
  - rOrtho\_1999-07-Spot4\_VS\_IMAGERY\_GCP\_opt\_loop\_#.csv: files containing the new coordinates corrected of each GCP and information used by the RSM or RFM refinement, by loop,
  - rOrtho\_1999-07-Spot4\_VS\_IMAGERY\_GCP\_optloop\_#\_correction.txt: files containing the correction for each loop,
  - RSM\_gcp\_patches: folder presenting all the patches for each GCP for each loop in the png and tif format,
  - SRTM\_DEM\_32636.vrt and SRTM\_DEM\_32636\_32635.vrt: files containing information on the DEM projected in a UTM grid.
- (b) **851a6814-ead2-11ed-a937-4cd98f389017\_IMAGERY**: this folder contains the same kind of results as (a) but that comes from the SPOT 2 image 003-006\_S2\_103-268-0\_1998-06-10-09-01-06\_HRV-1\_P\_DT\_TT.

- (c) **e5d7b1b0-eabf-11ed-a937-4cd98f389017\_IMAGERY**: this folder contains the same kind of results as (a) but that comes from the SPOT 2 image 003-008\_S2\_103-268-0\_1998-04-03-09-08-28\_HRV-1\_P\_DT\_TT.
  - (d) **geo\_ortho\_config.yaml**: this file presents all the functions with the different input parameters used to orthorectify image(s). This is the same file mentioned in 3.3.
  - (e) **input\_izmit\_correction\_model\_file.json**: this file presents the input parameters to get orthorectified images with a correction model file. This is the same file mentioned in 3.3.
  - (f) **RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW\_date\_hour.log**: this file presents the coding process. We have written the name of this file in input\_izmit.json, line “log\_name”, 3.3.
3. **output\_DEM**: folder containing the results from input parameters written in input\_izmit\_DEM.json. In output\_DEM we can find:
- (a) **6d8a8a6a-03d4-11ee-869f-4cd98f389017\_IMAGERY**: this folder contains:
    - IMAGERY\_ORTHO.tif: orthorectified image that comes from the SPOT 2 image 003-008\_S2\_103-268-0\_1998-04-03-09-08-28\_HRV-1\_P\_DT\_TT,
    - IMAGERY\_TRX.tif: this is the transformation matrix used for the orthorectification. This matrix contains Band-1 and Band-2 which correspond respectively to the 2D matrices of  $X(x_{pix}, y_{pix})$  and  $Y(x_{pix}, y_{pix})$  coordinates of the pixel  $p(x_{pix}, y_{pix})$  in the raw image to be projected (Aati et al., 2022),
    - SRTM\_DEM\_32635.vrt: information about the DEM.
  - (b) **33bcc6c2-03d4-11ee-869f-4cd98f389017\_IMAGERY**: this folder contains the same kind of results as (a) but that comes from the SPOT 2 image 003-008\_S2\_103-268-0\_1998-03-03-09-05-04\_HRV-1\_P\_DT\_TT,
  - (c) **fd5c8cc-03d3-11ee-869f-4cd98f389017\_IMAGERY**: this folder contains the same kind of results as (a) but that comes from the SPOT 2 image 003-006\_S2\_103-268-0\_1998-06-10-09-01-06\_HRV-1\_P\_DT\_TT,
  - (d) **geo\_ortho\_config.yaml**: this file presents all the functions with the different input parameters used to orthorectify image(s). This is the same file mentioned in 3.3.
  - (e) **input\_izmit\_DEM.json**: this file presents the input parameters to get orthorectified images by adding the elevation coordinates and without a reference image and a correction model file. This is the same file mentioned in 3.3.
  - (f) **RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW\_date\_hour.log**: this file presents the coding process. We have written the name of this file in input\_izmit.json, line “log\_name, 3.3”.
4. **output\_GCPs**: folder containing the results from input parameters written in input\_izmit\_GCPs.json. In output\_GCPs we can find:
- (a) **0b758192-eabe-11ed-a127-4cd98f389017\_IMAGERY**: this folder contains:
    - rOrtho\_1999-07-Spot4\_VS\_IMAGERY.pts: coordinates of each tie point (pixel location) between the raw image (Wrap Image) 003-008\_S2\_103-268-0\_1998-03-03-09-05-04\_HRV-1\_P\_DT\_TT and the reference image (Base Image),
    - rOrtho\_1999-07-Spot4\_VS\_IMAGER\_GCP.csv: coordinates of each GCP (projected in a geographic system) between the raw image 003-008\_S2\_103-268-0\_1998-03-03-09-05-04\_HRV-1\_P\_DT\_TT and the reference image.
  - (b) **4dad8f82-eabe-11ed-a127-4cd98f389017\_IMAGERY**: this folder contains the same kind of results as (a) but that comes from the SPOT 2 image 003-008\_S2\_103-268-0\_1998-04-03-09-08-28\_HRV-1\_P\_DT\_TT.
  - (c) **C65b8b24-eabd-11ed-a127-4cd98f389017\_IMAGERY**: this folder contains the same kind of results as (a) but that comes from the SPOT 2 image 003-006\_S2\_103-268-0\_1998-06-10-09-01-06\_HRV-1\_P\_DT\_TT.

- (d) **geo\_ortho\_config.yaml**: this file presents all the functions with the different input parameters used to orthorectify image(s). This is the same file mentioned in 3.3.
- (e) **input\_izmit\_GCPs.json**: this file presents the input parameters to get the tie points and GCPs (without the altitude coordinate) between the raw image(s) and a reference image. This is the same file mentioned in 3.3.
- (f) **RSM\_ORTHO\_SYSTEMATIC\_SP2\_WORKFLOW\_date\_hour.log**: this file presents the coding process. We have written the name of this file in input\_izmit.json, line “log\_name”, 3.3.

## 5. References

Aati, S., Milliner, C., Avouac, J.-P., 2022. A new approach for 2-D and 3-D precise measurements of ground deformation from optimized registration and correlation of optical images and ICA-based filtering of image geometry artifacts. California Institute of Technology, Pasadena, CA.