pyroSAR Documentation

Release 0.25

the pyroSAR Developers

Apr 16, 2024

CONTENTS

CHAPTER

ONE

GENERAL TOPICS

1.1 Installation

1.1.1 conda

Starting with version 0.11, pyroSAR is distributed via conda-forge and can easily be installed with

```
conda install --channel conda-forge pyrosar
```

This is by far the easiest way to work with pyroSAR on any operating system.

1.1.2 pip

Installation with pip is also supported and offers the advantage to install intermediate development stages directly from the GitHub repository. Mind however that several dependencies like GDAL cannot fully be installed this way. See further below for detailed Linux dependency installation instructions.

Installation of pip (Linux):

```
sudo apt-get install python-pip
```

The latest stable release of pyroSAR can then be installed:

```
python -m pip install pyroSAR
```

For installation of the latest master branch on GitHub, we need the version control system git. On Windows, git can be downloaded from git-scm.com. On Linux you can install it via command line:

```
sudo apt-get install git
```

Once everything is set up, pyroSAR is ready to be installed:

```
python -m pip install git+https://github.com/johntruckenbrodt/pyroSAR.git
```

1.1.3 Dependencies

The more specific instructions below are intended for Linux users who like to work outside of the Anaconda environment.

GDAL

pyroSAR requires GDAL version 2.1 with GEOS and PROJ4 as dependencies as well as the GDAL Python binding.

Ubuntu

Starting with release Yakkety (16.10), Ubuntu comes with GDAL >2.1. You can install it like this:

```
sudo apt-get install python-gdal python3-gdal gdal-bin
```

For older Ubuntu releases you can add the ubuntugis repository to apt prior to installation to install version >2.1:

```
sudo add-apt-repository ppa:ubuntugis/ppa
sudo apt-get update
```

This way the required dependencies (GEOS and PROJ4 in particular) are also installed. You can check the version by typing:

```
gdalinfo --version
```

Debian

Starting with Debian 9 (Stretch) GDAL is available in version >2.1 in the official repository.

Building from source

Alternatively, you can build GDAL and the dependencies from source. The script *pyroSAR/install/install_deps.sh* gives specific instructions on how to do it. It is not yet intended to run this script via shell, but rather to follow the instructions step by step.

SQLite + SpatiaLite

While *sqlite3* and its Python binding are usually already installed, the *spatialite* extension needs to be added. Two packages exist, *libspatialite* and *mod_spatialite*. Both can be used by pyroSAR. On Ubuntu, *mod_spatialite* has been found to be easier to setup with *sqlite* and can be installed via *apt*:

```
sudo apt-get install libsqlite3-mod-spatialite
```

On CentOS, *libspatialite* including shared objects for extension loading can be installed via *yum*:

```
sudo yum install libspatialite-devel
```

The following can be run in Python to test the needed functionality:

```
import sqlite3

# setup an in-memory database
con=sqlite3.connect(':memory:')

# enable loading extensions and load spatialite
con.enable_load_extension(True)
try:
    con.load_extension('mod_spatialite.so')
except sqlite3.OperationalError:
    con.load_extension('libspatialite.so')
```

In case loading extensions is not permitted you might need to install the package *pysqlite2* together with a static build of *sqlite3*. See the script *pyroSAR/install/install_deps.sh* for instructions. There you can also find instructions on how to install *spatialite* from source. To test *pysqlite2* you can import it as follows and then run the test above:

```
from pysqlite2 import dbapi2 as sqlite3
```

Installing this package is likely to cause problems with the *sqlite3* library installed on the system. Thus, it is safer to build a static *sqlite3* library for it (see installation script).

GAMMA

GAMMA's home directory as environment variable 'GAMMA_HOME' is expected to end either as GAMMA_SOFTWARE-<VERSIONNUMBER> or GAMMA_SOFTWARE/<VERSIONNUMBER>. If this differs in your install and cannot be changed, a workaround is adjusting the expected pattern in *ExamineGamma*.

1.2 File Naming

pyroSAR internally uses a fixed naming scheme to keep track of processed results. For each scene an identifier is created, which contains the sensor, acquisition mode, orbit (ascending or descending) and the time stamp of the acquisition start. For example *S1A__IW___A_20150222T170750*, which is created by calling method *outname_base()*:

```
from pyroSAR import identify
id = identify('S1A_IW_GRDH_1SDV_20150222T170750_20150222T170815_004739_005DD8_3768.zip
↪')
print(id.outname_base())
```

For each attribute a fixed number of digits is reserved. In case the attribute is shorter than this number, the rest of the digits is filled with underscores. I.e., the sensor field is four digits long, but 'S1A' only three. Thus, *S1A_* is the sensor slot. In the same way, *IW__* is the acquisition mode slot, which is also four digits long. *A* denotes ascending orbit, the time stamp is in format YYYYmmddTHHMMSS.

Processing functions like *geocode()* add suffixes to this identifier to further keep track of individual processing steps performed on the dataset. This core concept is used by many pyroSAR functions internally to keep track of which scenes have been processed before.

1.3 Handling of Orbit State Vector Files

SAR products require additional orbit state vector (OSV) information to improve their spatial location accuracy. This information is found in externally hosted files, which need to be downloaded separately and are then used by SAR processing software to update the product's metadata. Currently, pyroSAR only supports handling of Sentinel-1 OSV files.

In SNAP, the corresponding processing node is called *Apply-Orbit-File*, which automatically downloads the OSV file and updates the scene's metadata. The files are stored in SNAP's location for auxiliary data, which per default is *$HOME/.snap/auxdata/Orbits*.

In GAMMA, on the other hand, the downloading has to be done manually after which the command *isp.S1_OPOD_vec* can be used for updating the metadata. pyroSAR offers several approaches for automatically downloading these files. The central tool for managing existing files and downloading new ones is the class *pyroSAR.S1.OSV*, which is used for all approaches.

Note: in the following a dedicated directory is defined into which the files will be downloaded. If this directory is not defined (default is *None*), the files will be downloaded to SNAP's auxiliary data location (see above). This is

recommended as the files are kept in a central location that is accessible both by SNAP and by pyroSAR's GAMMA functionality.

1.3.1 approach 1: direct download by time span

In case a large number of scenes is to be processed and/or no internet access is available during processing, the files can be downloaded by time span to a central directory. This is the most basic approach using the central class *OSV* mentioned above, making use of its methods *catch()* and *retrieve()*.

```
from pyroSAR.S1 import OSV

osvdir = '/path/to/osvdir'

with OSV(osvdir) as osv:
    files = osv.catch(sensor='S1A', osvtype='POE',
                      start='20170101T000000', stop='20180101T000000',
                      url_option=1)
    osv.retrieve(files)
```

Two sub-directories *POEORB* and *RESORB* will be created in *osvdir* containing the downloaded files. *POEORB* will contain the *Precise Orbit Ephemerides* files, which are the most accurate but are first available about two weeks after the scene's acquisition. *RESORB* describes the *Restituted Orbit* files, which are less accurate but available directly after acquisition. See method *catch()* for download URL options.

1.3.2 approach 2: manual download per scene

The method *pyroSAR.drivers.SAFE.getOSV()* can be used to directly retrieve the files relevant for the scene. This method internally uses the methods described above with a time span limited to that of the scene acquisition.

```
from pyroSAR import identify
scene = 'S1A_IW_GRDH_1SDV_20180101T170648_20180101T170713_019964_021FFD_DA78.zip'
id = identify(scene)
match = id.getOSV(osvdir='/path/to/osvdir', osvType='POE', returnMatch=True)
print(match)
```

1.3.3 approach 3: direct download and scene metadata update (GAMMA only)

The convenience function *pyroSAR.gamma.correctOSV()* internally makes use of approach 2 and additionally directly executes the GAMMA command *isp.S1_OPOD_vec* for updating the scene's metadata with the information of the OSV file. The scene has to be unpacked first (see *pyroSAR.drivers.SAFE.unpack()*).

```
from pyroSAR import identify
from pyroSAR.gamma import correctOSV
scene = 'S1A_IW_GRDH_1SDV_20180101T170648_20180101T170713_019964_021FFD_DA78.zip'
id = identify(scene)
id.unpack('tmpdir')
correctOSV(id=id, osvdir='/path/to/osvdir', osvType='POE')
```

1.3.4 approach 4: automatic download and use during processing

The processing function *pyroSAR.gamma.geocode()* automatically downloads OSV files needed for processing and updates the scene's metadata using function *correctOSV()*. It is thus the most convenient way to handle these files and related processing steps. The parameter *allow_RES_OSV* can be used to allow processing with *RES* files if no *POE* file is available yet.

```
from pyroSAR.gamma import geocode
scene = 'S1A_IW_GRDH_1SDV_20180101T170648_20180101T170713_019964_021FFD_DA78.zip'
geocode(scene=scene,
        dem='/path/to/demfile',
        tmpdir='tmpdir',
        outdir='outdir',
        targetres=20,
        osvdir='/path/to/osvdir',
        allow_RES_OSV=False)
```

Similarly, the function *pyroSAR.snap.util.geocode()* also automatically downloads OSV files and chooses the best matching OSV type for processing.

```
from pyroSAR.snap import geocode
scene = 'S1A_IW_GRDH_1SDV_20180101T170648_20180101T170713_019964_021FFD_DA78.zip'
geocode(infile=scene,
        outdir='outdir',
        allow_RES_OSV=True)
```

In contrast to the GAMMA function, the OSV download directory cannot be set because of the fixed SNAP auxiliary data location. The type of the available OSV file is written to the workflow XML file for processing:

```
<node id="Apply-Orbit-File">
    <operator>Apply-Orbit-File</operator>
    <sources>
        <sourceProduct refid="Read"/>
    </sources>
    <parameters class="com.bc.ceres.binding.dom.XppDomElement">
        <orbitType>Sentinel Restituted (Auto Download)</orbitType>
        <polyDegree>3</polyDegree>
        <continueOnFail>false</continueOnFail>
    </parameters>
</node>
```

1.4 DEM Preparation

SAR processing requires a high resolution Digital Elevation Model for ortho-rectification and normalization of terrain-specific imaging effects.

In SNAP, the DEM is usually auto-downloaded by the software itself and the user only specifies the DEM source to be used, e.g. SRTM. pyroSAR's convenience function *pyroSAR.snap.util.geocode()* can additionally pass SNAP's option to use an external DEM file via parameters *externalDEMFile*, *externalDEMNoDataValue* and *externalDEMApplyEGM*.

GAMMA does not provide ways to automatically download DEMs for processing and the user thus also needs to provide an external DEM file in GAMMA's own format. However, several commands are available to prepare these DEMs including conversion from geoid heights to WGS84 ellipsoid heights.

pyroSAR offers several convenience functions to automatically prepare DEM mosaics from different sources to use them in either SNAP or GAMMA.

1.4.1 Download of DEM Tiles

The function *pyroSAR.auxdata.dem_autoload()* offers convenient download of tiles from different sources overlapping with user-defined geometries. Optionally, a buffer in degrees can be defined. This function internally makes use of the function `spatialist.auxil.gdalbuildvrt()`.

```
from pyroSAR.auxdata import dem_autoload
from spatialist import Vector

site = 'mysite.shp'
vrt = 'mosaic.vrt'

with Vector(site) as vec:
    vrt = dem_autoload(geometries=[vec],
                       demType='SRTM 1Sec HGT',
                       vrt=vrt,
                       buffer=0.1)
```

The tiles, which are delivered in compressed archives, are directly connected to a virtual mosaic using GDAL's VRT format, making it easier to work with them by treating them as a single file. For downloading tiles of some DEM types, e.g. *TDX90m*, an account needs to be created and the user credentials be passed to function *dem_autoload()*. See the function's documentation for further details.

The files are stored in SNAP's location for auxiliary data, which per default is *$HOME/.snap/auxdata/dem*. The function *dem_autoload()* has proven beneficial in server environments where not each node has internet access and the tiles thus need to be downloaded prior to processing on these nodes.

1.4.2 DEM Mosaicing

In a next step we create a mosaic GeoTIFF cropped to the boundaries defined in the VRT using the function *pyroSAR.auxdata.dem_create()*. The spatial reference system, WGS84 UTM 32N in this case, is defined by its EPSG code but also several other options are available. Since for SAR processing we are interested in ellipsoid heights, we call the function with the according parameter *geoid_convert* set to *True*. This function makes use of `spatialist.auxil.gdalwarp()`. Conversion of vertical reference systems, e.g. from geoid to ellipsoid, requires GDAL version >=2.2.

```
from pyroSAR.auxdata import dem_create

outname = 'mysite_srtm.tif'

dem_create(src=vrt, dst=outname,
           t_srs=32632, tr=(20, 20),
           resampling_method='bilinear',
           geoid_convert=True, geoid='EGM96')
```

1.4.3 GAMMA Import

For convenience, pyroSAR's *gamma* submodule contains a function *pyroSAR.gamma.dem.dem_autocreate()*, which is a combination of functions *dem_autoload()* and *dem_create()* and further executes GAMMA commands for format conversion. It offers the same parameters as these two functions and a user can additionally decide whether geoid-ellipsoid conversion is done in GDAL or in GAMMA via parameter *geoid_mode*. The output is a file in GAMMA format, which can directly be used for processing by e.g. function *pyroSAR.gamma.geocode()*.

1.5 SNAP API

pyroSAR offers a collection of tools to parse SNAP XML workflows and execute them with SNAP's Graph Processing Tool (GPT). All functionality is purely performed in Python and only the command line calls to GPT interact with SNAP. SNAP's Python API snappy is not used due to installation limitations and processing performance.

The following serves as a minimal example to showcase the core API functionality. A more complex example is given with function *pyroSAR.snap.util.geocode()*.

```
from pyroSAR.snap.auxil import parse_recipe, parse_node

workflow = parse_recipe('blank')

read = parse_node('Read')
read.parameters['file'] = 'S1A_IW_GRDH_1SDV_20150222T170750_20150222T170815_004739_
↪005DD8_3768.zip'
read.parameters['formatName'] = 'SENTINEL-1'
workflow.insert_node(read)

tnr = parse_node('ThermalNoiseRemoval')
workflow.insert_node(tnr, before=read.id)

bnr = parse_node('Remove-GRD-Border-Noise')
bnr.parameters['selectedPolarisations'] = ['VV']
workflow.insert_node(bnr, before=tnr.id)

write = parse_node('Write')
write.parameters['file'] = 'outname'
write.parameters['formatName'] = 'BEAM-DIMAP'
workflow.insert_node(write, before=bnr.id)

workflow.write('outname_proc')
```

Here, the function *parse_recipe()* is first used to create an empty workflow object of type *Workflow*. Using the function *parse_node()*, individual processing nodes can be loaded as *Node* objects and parameterized using a *Par* object via <node>.parameters. The method *insert_node()* is then used to insert the nodes into the workflow including linking of the nodes by modifying the source node entries. E.g. *Read* is set as source of the newly inserted *Remove-GRD-Border-Noise* node. As a last step, the workflow is written to an XML file with method *write()*.

This XML file can then be passed to function *gpt()* to process the workflow by internally calling the GPT command line tool:

```
from pyroSAR.snap.auxil import gpt

gpt('outname_proc.xml', tmpdir='.')
```

1.5.1 workflow splitting

Simple workflows like the one shown above take only a few seconds to process, but the more processing nodes are added, the more time it obviously takes to execute them. However, it was observed that executing long workflows takes longer and consumes more memory than executing each node individually. pyroSAR offers functionality to split long workflows into smaller groups and execute them in sequence with intermediate files being written in a temporary directory. First, the workflow nodes are grouped to contain a defined number of processing nodes, i.e. everything but *Read* and *Write*, using function *groupbyWorkers()*:

```
from pyroSAR.snap.auxil import groupbyWorkers

groupbyWorkers('outname_proc.xml', n=1)
```

This will return

```
[['Read', 'ThermalNoiseRemoval'], ['Remove-GRD-Border-Noise', 'Write']]
```

These groups can directly be passed passed to function *gpt()* via parameter `groups`. Internally the workflow is then split based on the groups and written to new XML files in a temporary directory using function *split()*. In this case, two workflows would be created:

- *Read -> ThermalNoiseRemoval -> Write*
- *Read -> Remove-GRD-Border-Noise -> Write*

These new files are then executed in sequence with intermediate *BEAM-DIMAP* files written in the same directory as the sub-workflow XML files. After processing this directory is deleted unless parameter `cleanup` of function *gpt()* is set to `False`.

1.5.2 backwards compatibility

With new versions of SNAP, new parameters are introduced and others removed. If a new parameter is not listed in the node's XML description its default is used by SNAP during processing. If, however, a parameter is contained in the workflow that is no longer supported by SNAP, the processing will be terminated. This can easily happen if the workflow was created by an older version of SNAP. pyroSAR reads the error messages and, if an unknown parameter is mentioned, deletes this parameter from the workflow, saves it to a new file and executes it instead.

1.5.3 troubleshooting

SNAP as well as pyroSAR's SNAP API are constantly being developed and bugs are unfortunately inevitable. This section is intended to guide users to better interpret errors and unexpected behaviour.

The process is running but seems inactive without any progress.

This might be related to SNAP's inability to download needed DEM tiles. SNAP will be stuck in a loop infinitely trying to download the missing tiles. This can be identified by directly running gpt in the command line. However, by operating gpt through a Python subprocess, it is not possible to see those command line messages. Only after a process has terminated, all messages can be retrieved and be written to log or error files.

A simple approach to interpret such a behaviour is to first create a workflow XML file with *geocode()*'s parameter `test=True` (so that only the XML is written but it is not executed):

```
from pyroSAR.snap import geocode
geocode(scene='S1A_IW_GRDH_1SDV_20200720T023849_20200720T023914_033532_03E2B5_2952.zip
↪',
        outdir='/test', test=True)
```

and then run gpt on it directly in the shell (i.e. outside of Python):

```
gpt /test/S1A__IW___D_20200720T023849_VV_Orb_ML_TC_proc.xml
```

This way one can directly see gpt's status, which in this case might be

```
SEVERE: org.esa.snap.core.dataop.dem.ElevationFile: java.lang.reflect.
↪InvocationTargetException
```

1.6 SAR Image Handling and Processing

1.6.1 Image Metadata

Let's start working with our actual satellite data. At first we load the scene into pyroSAR for analysis of the metadata:

```
from pyroSAR import identify
name = 'S1A_IW_GRDH_1SDV_20150222T170750_20150222T170815_004739_005DD8_3768.zip'
scene = identify(name)
print(scene)
```

This will automatically identify the scene, scan it for metadata and print a summary of selected metadata entries. Several attribute names (e.g. *sensor* and *acquisition_mode*) are standardized for all SAR scenes. Further entries, whose names are not standardized, can be found in a dictionary *scene.meta*. The function *identify()* will loop through all SAR images classes (*pyroSAR.drivers*) and return an object of the class that was successful in identifying the scene (*SAFE* in this case).

1.6.2 Database Handling

Now that we have made ourselves familiar with the scene, we can import its metadata into an SQLite database using class *Archive*:

```
from pyroSAR import Archive
dbfile = 'scenes.db'
with Archive(dbfile) as archive:
    archive.insert(scene)
```

dbfile is a file either containing an already existing database or one to be created. In this case an SQLite database with SpatiaLite extension is created. Alternatively, PostgreSQL + PostGIS can be used.

Let's assume our database contains a number of scenes and we want to select some for processing. We have a shapefile, which contains a geometry delimiting our test site for which we want to process some Sentinel-1 scenes. We already processed some scenes in the past and the results are stored in a directory *outdir*. We only want to select scenes which have not been processed to this directory before. Furthermore, we are only interested in scenes acquired in Ground Range Detected (GRD) Interferometric Wide Swath mode (IW), which contain a VV band.

```
from spatialist import Vector
archive = Archive('scenes.db')
outdir = '/path/to/processed/results'
maxdate = '20171231T235959'
with Vector('site.shp') as site:
    selection_proc = archive.select(vectorobject=site,
                                    processdir=outdir,
                                    maxdate=maxdate,
                                    sensor=('S1A', 'S1B'),
                                    product='GRD',
                                    acquisition_mode='IW',
                                    vv=1)
archive.close()
```

Here we use the vector geometry driver of package spatialist, which is developed alongside of pyroSAR. The `spatialist.Vector` object is then passed to method *Archive.select*.

1.6.3 Processing

The returned *selection_proc* is a list of file names for the scenes we selected from the database, which we can now pass to a processing function:

```
from pyroSAR.snap import geocode

# the target pixel spacing in meters
spacing = 20

for scene in selection_proc:
    geocode(infile=scene, outdir=outdir, tr=spacing, scaling='db', shapefile=site)
```

The function *snap.geocode* is a basic utility for SNAP. It will perform all necessary steps to subset, resample, topographically normalize, geocode and scale the input image and write GeoTIFF files to the selected output directory. All necessary files like orbit state vectors and SRTM DEM tiles are downloaded automatically in the background by SNAP. SNAP is most conveniently used with workflow XMLs. The function geocode parses a workflow for the particular scene, parametrizes it (depending on the scene type and selected processing parameters) and writes it to the output directory. It then calls the command *gpt*, which is SNAP's command line interface, on the workflow to execute the processing steps.

1.7 Logging

pyroSAR makes use of the `logging` module to display status messages for running processes. See Logging HOWTO for a basic tutorial. To display log messages you may add one of the following examples to your script:

```
import logging

# basic info
logging.basicConfig(level=logging.INFO)

# basic info with some message filtering
logging.basicConfig(format='%(levelname)s:%(message)s', level=logging.INFO)

# detailed debug info
logging.basicConfig(level=logging.DEBUG)
```

CHAPTER

TWO

API DOCUMENTATION

2.1 Drivers

This is the core module of package pyroSAR. It contains the drivers for the different SAR image formats and offers functionality for retrieving metadata, unpacking images, downloading ancillary files like DEMs and Orbit State Vector files as well as archiving scenes in a database. The *ID* class and its subclasses allow easy and standardized access to the metadata of images from different SAR sensors.

classes

ID	Abstract class for SAR meta data handlers
BEAM_DIMAP	Handler class for BEAM-DIMAP data
CEOS_PSR	Handler class for ALOS-PALSAR data in CEOS format
CEOS_ERS	Handler class for ERS data in CEOS format
EORC_PSR	Handler class for ALOS-2/PALSAR-2 data in EORC (Earth Observation Research Center) Path format
ESA	Handler class for SAR data in ESA format (Envisat ASAR, ERS-1/2)
SAFE	Handler class for Sentinel-1 data
TSX	Handler class for TerraSAR-X and TanDEM-X data
TDM	Handler class for TerraSAR-X and TanDEM-X experimental data
Archive	Utility for storing SAR image metadata in a database

functions

identify	identify a SAR scene and return the appropriate metadata handler object
identify_many	wrapper function for returning metadata handlers of all valid scenes in a list, similar to function *identify()*.
filter_processed	Filter a list of pyroSAR objects to those that have not yet been processed and stored in the defined directory.
getFileObj	Load a file in a SAR scene archive into a readable file object.
parse_date	this function gathers known time formats provided in the different SAR products and converts them to a common standard of the form YYYYMMDDTHHMMSS
drop_archive	drop (delete) a scene database

class pyroSAR.drivers.**Archive**(*dbfile*, *custom_fields=None*, *postgres=False*, *user='postgres'*, *password='1234'*, *host='localhost'*, *port=5432*, *cleanup=True*, *legacy=False*)

Bases: object

Utility for storing SAR image metadata in a database

Parameters

- **dbfile** (*str*) – the filename for the SpatiaLite database. This might either point to an existing database or will be created otherwise. If postgres is set to True, this will be the name for the PostgreSQL database.
- **custom_fields** (*dict or None*) – a dictionary containing additional non-standard database column names and data types; the names must be attributes of the SAR scenes to be inserted (i.e. id.attr) or keys in their meta attribute (i.e. id.meta['attr'])
- **postgres** (*bool*) – enable postgres driver for the database. Default: False
- **user** (*str*) – required for postgres driver: username to access the database. Default: 'postgres'
- **password** (*str*) – required for postgres driver: password to access the database. Default: '1234'
- **host** (*str*) – required for postgres driver: host where the database is hosted. Default: 'localhost'
- **port** (*int*) – required for postgres driver: port number to the database. Default: 5432
- **cleanup** (*bool*) – check whether all registered scenes exist and remove missing entries?
- **legacy** (*bool*) – open an outdated database in legacy mode to import into a new database. Opening an outdated database without legacy mode will throw a RuntimeError.

Examples

Ingest all Sentinel-1 scenes in a directory and its sub-directories into the database:

```
>>> from pyroSAR import Archive, identify
>>> from spatialist.ancillary import finder
>>> dbfile = '/.../scenelist.db'
>>> archive_s1 = '/.../sentinel1/GRD'
>>> scenes_s1 = finder(archive_s1, [r'^S1[AB].*\.zip'], regex=True,␣
↪recursive=True)
>>> with Archive(dbfile) as archive:
>>>     archive.insert(scenes_s1)
```

select all Sentinel-1 A/B scenes stored in the database, which

- overlap with a test site
- were acquired in Ground-Range-Detected (GRD) Interferometric Wide Swath (IW) mode before 2018
- contain a VV polarization image
- have not been processed to directory *outdir* before

```
>>> from pyroSAR import Archive
>>> from spatialist import Vector
>>> archive = Archive('/.../scenelist.db')
>>> site = Vector('/path/to/site.shp')
>>> outdir = '/path/to/processed/results'
```

(continues on next page)

(continued from previous page)

```
>>> maxdate = '20171231T235959'
>>> selection_proc = archive.select(vectorobject=site, processdir=outdir,
>>>                                 maxdate=maxdate, sensor=('S1A', 'S1B'),
>>>                                 product='GRD', acquisition_mode='IW', vv=1)
>>> archive.close()
```

Alternatively, the *with* statement can be used. In this case to just check whether one particular scene is already registered in the database:

```
>>> from pyroSAR import identify, Archive
>>> scene = identify('S1A_IW_SLC__1SDV_20150330T170734_20150330T170801_005264_
↪006A6C_DA69.zip')
>>> with Archive('/.../scenelist.db') as archive:
>>>     print(archive.is_registered(scene.scene))
```

When providing 'postgres' as driver, a PostgreSQL database will be created at a given host. Additional arguments are required.

```
>>> from pyroSAR import Archive, identify
>>> from spatialist.ancillary import finder
>>> dbfile = 'scenelist_db'
>>> archive_s1 = '/.../sentinel1/GRD'
>>> scenes_s1 = finder(archive_s1, [r'^S1[AB].*\.zip'], regex=True,␣
↪recursive=True)
>>> with Archive(dbfile, driver='postgres', user='user', password='password',␣
↪host='host', port=5432) as archive:
>>>     archive.insert(scenes_s1)
```

Importing an old database:

```
>>> from pyroSAR import Archive
>>> db_new = 'scenes.db'
>>> db_old = 'scenes_old.db'
>>> with Archive(db_new) as db:
>>>     with Archive(db_old, legacy=True) as db_old:
>>>         db.import_outdated(db_old)
```

add_tables(*tables*)

Add tables to the database per `sqlalchemy.schema.Table` Tables provided here will be added to the database.

Note: Columns using Geometry must have setting management=True for SQLite, for example: `geometry = Column(Geometry('POLYGON', management=True, srid=4326))`

Parameters

tables (`sqlalchemy.schema.Table` or list[`sqlalchemy.schema.Table`]) – The table(s) to be added to the database.

cleanup()

Remove all scenes from the database, which are no longer stored in their registered location

close()

close the database connection

drop_element(*scene*, *with_duplicates=False*)

Drop a scene from the data table. If duplicates table contains matching entry, it will be moved to the data table.

Parameters

- **scene** (*str*) – a SAR scene
- **with_duplicates** (*bool*) – True: delete matching entry in duplicates table False: move matching entry from duplicates into data table

drop_table(*table*)

Drop a table from the database.

Parameters
table (*str*) – the table name

static encode(*string*, *encoding='utf-8'*)

export2shp(*path*, *table='data'*)

export the database to a shapefile

Parameters

- **path** (*str*) – the path of the shapefile to be written. This will overwrite other files with the same name. If a folder is given in path it is created if not existing. If the file extension is missing '.shp' is added.
- **table** (*str*) – the table to write to the shapefile; either 'data' (default) or 'duplicates'

filter_scenelist(*scenelist*)

Filter a list of scenes by file names already registered in the database.

Parameters
scenelist (*list[str or ID]*) – the scenes to be filtered

Returns
the file names of the scenes whose basename is not yet registered in the database

Return type
list[*ID*]

get_colnames(*table='data'*)

Return the names of all columns of a table.

Returns
the column names of the chosen table

Return type
list[str]

get_tablenames(*return_all=False*)

Return the names of all tables in the database

Parameters
return_all (*bool*) – only gives tables data and duplicates on default. Set to True to get all other tables and views created automatically.

Returns
the table names

Return type
list[str]

get_unique_directories()

Get a list of directories containing registered scenes

Returns
the directory names

Return type
list[str]

import_outdated(*dbfile*)

import an older database

Parameters
dbfile (*str or Archive*) – the old database. If this is a string, the name of a CSV file is expected.

insert(*scene_in*, *pbar=False*, *test=False*)

Insert one or many scenes into the database

Parameters

- **scene_in** (*str or ID or list[str or ID]*) – a SAR scene or a list of scenes to be inserted
- **pbar** (*bool*) – show a progress bar?
- **test** (*bool*) – should the insertion only be tested or directly be committed to the database?

is_registered(*scene*)

Simple check if a scene is already registered in the database.

Parameters
scene (*str or ID*) – the SAR scene

Returns
is the scene already registered?

Return type
bool

move(*scenelist*, *directory*, *pbar=False*)

Move a list of files while keeping the database entries up to date. If a scene is registered in the database (in either the data or duplicates table), the scene entry is directly changed to the new location.

Parameters

- **scenelist** (*list[str]*) – the file locations
- **directory** (*str*) – a folder to which the files are moved
- **pbar** (*bool*) – show a progress bar?

select(*vectorobject=None*, *mindate=None*, *maxdate=None*, *date_strict=True*, *processdir=None*, *recursive=False*, *polarizations=None*, ***args*)

select scenes from the database

Parameters

- **vectorobject** (Vector or None) – a geometry with which the scenes need to overlap
- **mindate** (*str or datetime.datetime or None*) – the minimum acquisition date; strings must be in format YYYYmmddTHHMMSS; default: None
- **maxdate** (*str or datetime.datetime or None*) – the maximum acquisition date; strings must be in format YYYYmmddTHHMMSS; default: None
- **date_strict** (*bool*) – treat dates as strict limits or also allow flexible limits to incorporate scenes whose acquisition period overlaps with the defined limit?
  - strict: start >= mindate & stop <= maxdate

– not strict: stop >= mindate & start <= maxdate

- **processdir** (*str or None*) – A directory to be scanned for already processed scenes; the selected scenes will be filtered to those that have not yet been processed. Default: None
- **recursive** (*bool*) – (only if *processdir* is not None) should also the subdirectories of the *processdir* be scanned?
- **polarizations** (*list[str] or None*) – a list of polarization strings, e.g. ['HH', 'VV']
- ****args** – any further arguments (columns), which are registered in the database. See *get_colnames()*

Returns
the file names pointing to the selected scenes

Return type
list[str]

select_duplicates(*outname_base=None*, *scene=None*, *value='id'*)

Select scenes from the duplicates table. In case both *outname_base* and *scene* are set to None all scenes in the table are returned, otherwise only those that match the attributes *outname_base* and *scene* if they are not None.

Parameters

- **outname_base** (*str*) – the basename of the scene
- **scene** (*str*) – the scene name
- **value** (*str*) – the return value; either 'id' or 'scene'

Returns
the selected scene(s)

Return type
list[str]

property size

get the number of scenes registered in the database

Returns
the number of scenes in (1) the main table and (2) the duplicates table

Return type
tuple[int]

class pyroSAR.drivers.**BEAM_DIMAP**(*scene*)

Bases: *ID*

Handler class for BEAM-DIMAP data

Sensors:

- SNAP supported sensors

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

class pyroSAR.drivers.**CEOS_ERS**(*scene*)

Bases: *ID*

Handler class for ERS data in CEOS format

Sensors:

- ERS1
- ERS2

Reference:

ER-IS-EPO-GS-5902-3: Annex C. ERS SAR.SLC/SLC-I. CCT and EXABYTE (ESA 1998)

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns

the derived attributes

Return type

dict

unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

class pyroSAR.drivers.**CEOS_PSR**(*scene*)

Bases: *ID*

Handler class for ALOS-PALSAR data in CEOS format

Sensors:

- PSR1
- PSR2

PALSAR-1:

References:

- NEB-01006: ALOS/PALSAR Level 1 Product Format Description (JAXA 2006)
- NEB-070062B: ALOS/PALSAR Level 1.1/1.5 Product Format Description (JAXA 2009)

Products / processing levels:

- 1.0
- 1.1

- 1.5

Acquisition modes:

- AB: [SP][HWDPC]
- **A: supplemental remarks of the sensor type:**
  - S: Wide observation mode
  - P: all other modes
- **B: observation mode**
  - H: Fine mode
  - W: ScanSAR mode
  - D: Direct downlink mode
  - P: Polarimetry mode
  - C: Calibration mode

PALSAR-2:

Reference:
ALOS-2/PALSAR-2 Level 1.1/1.5/2.1/3.1 CEOS SAR Product Format Description (JAXA 2014).

Products / processing levels:

- 1.0
- 1.1
- 1.5

Acquisition modes:

- SBS: Spotlight mode
- UBS: Ultra-fine mode Single polarization
- UBD: Ultra-fine mode Dual polarization
- HBS: High-sensitive mode Single polarization
- HBD: High-sensitive mode Dual polarization
- HBQ: High-sensitive mode Full (Quad.) polarimetry
- FBS: Fine mode Single polarization
- FBD: Fine mode Dual polarization
- FBQ: Fine mode Full (Quad.) polarimetry
- WBS: Scan SAR nominal [14MHz] mode Single polarization
- WBD: Scan SAR nominal [14MHz] mode Dual polarization
- WWS: Scan SAR nominal [28MHz] mode Single polarization
- WWD: Scan SAR nominal [28MHz] mode Dual polarization
- VBS: Scan SAR wide mode Single polarization
- VBD: Scan SAR wide mode Dual polarization

`property led_filename`

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

class pyroSAR.drivers.**EORC_PSR**(*scene*)

Bases: *ID*

Handler class for ALOS-2/PALSAR-2 data in EORC (Earth Observation Research Center) Path format

Sensors:

- PALSAR-2

PALSAR-2:

Reference:
NDX-150019: ALOS-2/PALSAR-2 EORC Path Product Format Description (JAXA 2016)

Products / processing levels:

- 1.5

Acquisition modes:

- FBD: Fine mode Dual polarization
- WBD: Scan SAR nominal [14MHz] mode Dual polarization

property header_filename

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

class pyroSAR.drivers.**ESA**(*scene*)

Bases: *ID*

Handler class for SAR data in ESA format (Envisat ASAR, ERS-1/2)

Sensors:

- ASAR
- ERS1
- ERS2

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

class pyroSAR.drivers.**ID**(*metadict*)

Bases: object

Abstract class for SAR meta data handlers

bbox(*outname=None*, *driver=None*, *overwrite=True*)

get the bounding box of a scene either as a vector object or written to a file

Parameters

- **outname** (*str*) – the name of the vector file to be written
- **driver** (*str*) – the output file format; needs to be defined if the format cannot be auto-detected from the filename extension
- **overwrite** (*bool*) – overwrite an existing vector file?

Returns
the vector object if *outname* is None and None otherwise

Return type
Vector or None

See also:

spatialist.vector.Vector.bbox

property compression

check whether a scene is compressed into an tarfile or zipfile or not at all

Returns
either 'zip', 'tar' or None

Return type
str or None

examine(*include_folders=False*)

check whether any items in the SAR scene structure (i.e. files/folders) match the regular expression pattern defined by the class. On success the item is registered in the object as attribute *file*.

Parameters
include_folders (*bool*) – also match folder (or just files)?

Raises
RuntimeError –

export2dict()

Return the uuid and the metadata that is defined in *self.locals* as a dictionary

export2sqlite(*dbfile*)

Export relevant metadata to an SQLite database

Parameters
dbfile (*str*) – the database file

findfiles(*pattern*, *include_folders=False*)

find files in the scene archive, which match a pattern.

Parameters

- **pattern** (*str*) – the regular expression to match
- **include_folders** (*bool*) – also match folders (or just files)?

Returns
the matched file names

Return type
list[str]

See also:

spatialist.ancillary.finder()

gdalinfo()

read metadata directly from the GDAL SAR image drivers

Returns
the metadata attributes

Return type
dict

geometry(*outname=None*, *driver=None*, *overwrite=True*)

get the footprint geometry of a scene either as a vector object or written to a file

Parameters

- **outname** (*str*) – the name of the vector file to be written
- **driver** (*str*) – the output file format; needs to be defined if the format cannot be auto-detected from the filename extension
- **overwrite** (*bool*) – overwrite an existing vector file?

Returns
the vector object if *outname* is None, None otherwise

Return type
Vector or None

See also:

spatialist.vector.Vector.write

`getCorners`()

Get the bounding box corner coordinates

Returns
the corner coordinates as a dictionary with keys *xmin*, *ymin*, *xmax*, *ymax*

Return type
dict

`getFileObj`(*filename*)

Load a file into a readable file object.

Parameters
filename (*str*) – the name of a file in the scene archive, easiest to get with method *findfiles()*

Returns
a file pointer object

Return type
io.BytesIO

`getGammaImages`(*directory=None*)

list all files processed by GAMMA

Parameters
directory (*str or None*) – the directory to be scanned; if left empty the object attribute *gammadir* is scanned

Returns
the file names of the images processed by GAMMA

Return type
list[str]

Raises
RuntimeError –

`getHGT`()

get the names of all SRTM HGT tiles overlapping with the SAR scene

Returns
names of the SRTM HGT tiles

Return type
list[str]

`is_processed`(*outdir*, *recursive=False*)

check whether a scene has already been processed and stored in the defined output directory (and subdirectories if scanned recursively)

Parameters
outdir (*str*) – the directory to be checked

Returns
does an image matching the scene pattern exist?

Return type
bool

`outname_base`(*extensions=None*)

parse a string containing basic information about the scene in standardized format. Currently, this id contains the sensor (4 digits), acquisition mode (4 digits), orbit (1 digit) and acquisition start time (15 digits)., e.g. *S1A__IW___A_20150523T122350*.

Parameters
extensions (*list[str]*) – the names of additional parameters to append to the basename, e.g. `['orbitNumber_rel']`

Returns
a standardized name unique to the scene

Return type
str

static parse_date(*x*)

this function gathers known time formats provided in the different SAR products and converts them to a common standard of the form YYYYMMDDTHHMMSS.

Parameters
x (*str*) – the time stamp

Returns
the converted time stamp in format YYYYmmddTHHMMSS

Return type
str

abstract quicklook(*outname*, *format='kmz'*)

export a quick look image of the scene

Parameters

- **outname** (*str*) – the name of the output file
- **format** (*str*) – the format of the file to write; currently only kmz is supported

Examples

```
>>> from pyroSAR import identify
>>> scene = identify('S1A_IW_GRDH_1SDV_20180101T170648_20180101T170713_
↪019964_021FFD_DA78.zip')
>>> scene.quicklook('S1A__IW___A_20180101T170648.kmz')
```

abstract scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

summary()

print the set of standardized scene metadata attributes

abstract unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

class pyroSAR.drivers.**SAFE**(*scene*)

Bases: *ID*

Handler class for Sentinel-1 data

Sensors:

- S1A
- S1B

References:

- S1-RS-MDA-52-7443 Sentinel-1 IPF Auxiliary Product Specification
- MPC-0243 Masking "No-value" Pixels on GRD Products generated by the Sentinel-1 ESA IPF

getOSV(*osvdir=None*, *osvType='POE'*, *returnMatch=False*, *useLocal=True*, *timeout=300*, *url_option=1*)

download Orbit State Vector files for the scene

Parameters

- **osvdir** (*str*) – the directory of OSV files; subdirectories POEORB and RESORB are created automatically; if no directory is defined, the standard SNAP auxdata location is used
- **osvType** (*str or list[str]*) – the type of orbit file either 'POE', 'RES' or a list of both; if both are selected, the best matching file will be retrieved. I.e., POE if available and RES otherwise
- **returnMatch** (*bool*) – return the best matching orbit file?
- **useLocal** (*bool*) – use locally existing files and do not search for files online if the right file has been found?
- **timeout** (*int or tuple or None*) – the timeout in seconds for downloading OSV files as provided to `requests.get()`
- **url_option** (*int*) – the OSV download URL option; see *pyroSAR.S1.OSV.catch()* for options

Returns

the best matching OSV file if *returnMatch* is True or None otherwise

Return type

str or None

See also:

pyroSAR.S1.OSV

quicklook(*outname*, *format='kmz'*, *na_transparent=True*)

Write a quicklook file for the scene.

Parameters

- **outname** (*str*) – the file to write
- **format** (*str*) – the quicklook format. Currently supported options:
  - kmz
- **na_transparent** (*bool*) – make NA values transparent?

removeGRDBorderNoise(*method='pyroSAR'*)

mask out Sentinel-1 image border noise.

Parameters

method (*str*) – the border noise removal method to be applied; one of the following:

- 'ESA': the pure implementation as described by ESA

- 'pyroSAR': the ESA method plus the custom pyroSAR refinement

See also:

removeGRDBorderNoise()

resolution()

Compute the mid-swath resolution of the Sentinel-1 product. For GRD products the resolution is expressed in ground range and in slant range otherwise.

References:

- https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/resolutions/level-1-single-look-complex
- https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/resolutions/level-1-ground-range-detected
- https://sentinel.esa.int/web/sentinel/user-guides/sentinel-1-sar/document-library/-/asset_publisher/1dO7RF5fJMbd/content/sentinel-1-product-definition

Returns
the resolution as (range, azimuth)

Return type
tuple[float]

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

class pyroSAR.drivers.**TDM**(*scene*)

Bases: *TSX*

Handler class for TerraSAR-X and TanDEM-X experimental data

Sensors:

- TDM1

References:

- TD-GS-PS-3028 TanDEM-X Experimental Product Description

Acquisition modes:

- HS: High Resolution SpotLight
- SL: SpotLight
- SM: StripMap

Polarisation modes:

- Single (S): all acquisition modes
- Dual (D): High Resolution SpotLight (HS), SpotLight (SL) and StripMap (SM)
- Twin (T): StripMap (SM) (experimental)
- Quad (Q): StripMap (SM) (experimental)

Products:

- CoSSCs: (bi-static) SAR co-registered single look slant range complex products (CoSSCs)

Examples

Ingest all Tandem-X Bistatic scenes in a directory and its sub-directories into the database:

```
>>> from pyroSAR import Archive, identify
>>> from spatialist.ancillary import finder
>>> dbfile = '/.../scenelist.db'
>>> archive_tdm = '/.../TDM/'
>>> scenes_tdm = finder(archive_tdm, [r'^TDM1.*'], foldermode=2, regex=True,␣
↪recursive=True)
>>> with Archive(dbfile) as archive:
>>>     archive.insert(scenes_tdm)
```

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

class pyroSAR.drivers.**TSX**(*scene*)

Bases: *ID*

Handler class for TerraSAR-X and TanDEM-X data

Sensors:

- TSX1
- TDX1

References:

- TX-GS-DD-3302 TerraSAR-X Basic Product Specification Document
- TX-GS-DD-3303 TerraSAR-X Experimental Product Description
- TD-GS-PS-3028 TanDEM-X Experimental Product Description
- TerraSAR-X Image Product Guide (Airbus Defence and Space)

Acquisition modes:

- ST: Staring Spotlight
- HS: High Resolution SpotLight
- HS300: High Resolution SpotLight 300 MHz
- SL: SpotLight

- SM: StripMap
- SC: ScanSAR
- WS: Wide ScanSAR

Polarisation modes:

- Single (S): all acquisition modes
- Dual (D): High Resolution SpotLight (HS), SpotLight (SL) and StripMap (SM)
- Twin (T): StripMap (SM) (experimental)
- Quad (Q): StripMap (SM) (experimental)

Products:

- SSC: Single Look Slant Range Complex
- MGD: Multi Look Ground Range Detected
- GEC: Geocoded Ellipsoid Corrected
- EEC: Enhanced Ellipsoid Corrected

scanMetadata()

scan SAR scenes for metadata attributes. The returned dictionary is registered as attribute *meta* by the class upon object initialization. This dictionary furthermore needs to return a set of standardized attribute keys, which are directly registered as object attributes.

Returns
the derived attributes

Return type
dict

unpack(*directory*, *overwrite=False*, *exist_ok=False*)

Unpack the SAR scene into a defined directory.

Parameters

- **directory** (*str*) – the base directory into which the scene is unpacked
- **overwrite** (*bool*) – overwrite an existing unpacked scene?
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?

pyroSAR.drivers.**drop_archive**(*archive*)

drop (delete) a scene database

Parameters
archive (pyroSAR.drivers.Archive) – the database to be deleted

See also:

sqlalchemy_utils.functions.drop_database()

Examples

```
>>> pguser = os.environ.get('PGUSER')
>>> pgpassword = os.environ.get('PGPASSWORD')
```

```
>>> db = Archive('test', postgres=True, port=5432, user=pguser,␣
↪password=pgpassword)
>>> drop_archive(db)
```

pyroSAR.drivers.**filter_processed**(*scenelist*, *outdir*, *recursive=False*)

Filter a list of pyroSAR objects to those that have not yet been processed and stored in the defined directory. The search for processed scenes is either done in the directory only or recursively into subdirectories. The scenes must have been processed with pyroSAR in order to follow the right naming scheme.

Parameters

- **scenelist** (*list[ID]*) – a list of pyroSAR objects
- **outdir** (*str*) – the processing directory
- **recursive** (*bool*) – scan *outdir* recursively into subdirectories?

Returns
a list of those scenes, which have not been processed yet

Return type
list[*ID*]

pyroSAR.drivers.**getFileObj**(*scene*, *filename*)

Load a file in a SAR scene archive into a readable file object.

Parameters

- **scene** (*str*) – the scene archive. Can be either a directory or a compressed archive of type *zip* or *tar.gz*.
- **filename** (*str*) – the name of a file in the scene archive, easiest to get with method *findfiles()*

Returns
a file object

Return type
BytesIO

pyroSAR.drivers.**identify**(*scene*)

identify a SAR scene and return the appropriate metadata handler object

Parameters
scene (*str*) – a file or directory name

Returns
a pyroSAR metadata handler

Return type
pyroSAR.drivers.ID

Examples

```
>>> from pyroSAR import identify
>>> filename = 'S1A_IW_GRDH_1SDV_20180829T170656_20180829T170721_023464_028DE0_
↪F7BD.zip'
>>> scene = identify(filename)
>>> print(scene)
pyroSAR ID object of type SAFE
acquisition_mode: IW
cycleNumber: 148
frameNumber: 167392
lines: 16703
orbit: A
orbitNumber_abs: 23464
orbitNumber_rel: 117
polarizations: ['VV', 'VH']
```

(continues on next page)

(continued from previous page)

```
product: GRD
projection: +proj=longlat +datum=WGS84 +no_defs
samples: 26056
sensor: S1A
spacing: (10.0, 10.0)
start: 20180829T170656
stop: 20180829T170721
```

pyroSAR.drivers.**identify_many**(*scenes*, *pbar=False*, *sortkey=None*)

wrapper function for returning metadata handlers of all valid scenes in a list, similar to function *identify()*.

Parameters

- **scenes** (*list[str or ID]*) – the file names of the scenes to be identified
- **pbar** (*bool*) – adds a progressbar if True
- **sortkey** (*str or None*) – sort the handler object list by an attribute

Returns

a list of pyroSAR metadata handlers

Return type

list[*ID*]

Examples

```
>>> from pyroSAR import identify_many
>>> files = finder('/path', ['S1*.zip'])
>>> ids = identify_many(files, pbar=False, sortkey='start')
```

pyroSAR.drivers.**parse_date**(*x*)

this function gathers known time formats provided in the different SAR products and converts them to a common standard of the form YYYYMMDDTHHMMSS

Parameters

x (*str or datetime*) – the time stamp to be converted

Returns

the converted time stamp in format YYYYmmddTHHMMSS

Return type

str

2.2 SNAP

2.2.1 Processing

geocode	general function for geocoding of SAR backscatter images with SNAP.
noise_power	Generate Sentinel-1 noise power images for each polarization, calibrated to either beta, sigma or gamma nought.

pyroSAR.snap.util.**geocode**(*infile*, *outdir*, *t_srs=4326*, *spacing=20*, *polarizations='all'*, *shapefile=None*, *scaling='dB'*, *geocoding_type='Range-Doppler'*, *removeS1BorderNoise=True*, *removeS1BorderNoiseMethod='pyroSAR'*, *removeS1ThermalNoise=True*, *offset=None*, *allow_RES_OSV=False*, *demName='SRTM 1Sec HGT'*, *externalDEMFile=None*, *externalDEMNoDataValue=None*, *externalDEMApplyEGM=True*, *terrainFlattening=True*, *basename_extensions=None*, *test=False*, *export_extra=None*, *groupsize=1*, *cleanup=True*, *tmpdir=None*, *gpt_exceptions=None*, *gpt_args=None*, *returnWF=False*, *nodataValueAtSea=True*, *demResamplingMethod='BILINEAR_INTERPOLATION'*, *imgResamplingMethod='BILINEAR_INTERPOLATION'*, *alignToStandardGrid=False*, *standardGridOriginX=0*, *standardGridOriginY=0*, *speckleFilter=False*, *refarea='gamma0'*, *clean_edges=False*, *clean_edges_npixels=1*, *rlks=None*, *azlks=None*, *dem_oversampling_multiple=2*, *s1_osv_url_option=1*)

general function for geocoding of SAR backscatter images with SNAP.

This function performs the following steps:

- (if necessary) identify the SAR scene(s) passed via argument *infile* (*pyroSAR.drivers.identify()*)
- (if necessary) create the directories defined via *outdir* and *tmpdir*
- (if necessary) download Sentinel-1 OSV files
- parse a SNAP workflow (*pyroSAR.snap.auxil.Workflow*)
- write the workflow to an XML file in *outdir*
- execute the workflow (*pyroSAR.snap.auxil.gpt()*)

Note: The function may create workflows with multiple *Write* nodes. All nodes are parametrized to write data in ENVI format, in which case the node parameter *file* is going to be a directory. All nodes will use the same temporary directory, which will be created in *tmpdir*. Its name is created from the basename of the *infile* (*pyroSAR.drivers.ID.outname_base()*) and a suffix identifying each processing node of the workflow (*pyroSAR.snap.auxil.Workflow.suffix()*).

For example: *S1A__IW___A_20180101T170648_NR_Orb_Cal_ML_TF_TC*.

Parameters

- **infile** (*str or ID or list*) – The SAR scene(s) to be processed; multiple scenes are treated as consecutive acquisitions, which will be mosaicked with SNAP's Slice-Assembly operator.
- **outdir** (*str*) – The directory to write the final files to.
- **t_srs** (*int or str or osgeo.osr.SpatialReference*) – A target spatial reference system in WKT, EPSG, PROJ4 or OPENGIS format. See function `spatialist.auxil.crsConvert()` for details. Default: 4326.
- **spacing** (*int or float, optional*) – The target pixel spacing in meters. Default is 20
- **polarizations** (*list[str] or str*) – The polarizations to be processed; can be a string for a single polarization, e.g. 'VV', or a list of several polarizations, e.g. ['VV', 'VH']. With the special value 'all' (default) all available polarizations are processed.
- **shapefile** (str or `Vector` or dict, optional) – A vector geometry for spatial subsetting:
  - `Vector`: a vector object in arbitrary CRS
  - `str`: a name of a file that can be read with `Vector` in arbitrary CRS
  - `dict`: a dictionary with keys *xmin*, *xmax*, *ymin*, *ymax* in EPSG:4326 coordinates

- **scaling** (*{'dB', 'db', 'linear'}, optional*) – Should the output be in linear or decibel scaling? Default is 'dB'.
- **geocoding_type** (*{'Range-Doppler', 'SAR simulation cross correlation'}, optional*) – The type of geocoding applied; can be either 'Range-Doppler' (default) or 'SAR simulation cross correlation'
- **removeS1BorderNoise** (*bool, optional*) – Enables removal of S1 GRD border noise (default). Will be ignored if SLC scenes are processed.
- **removeS1BorderNoiseMethod** (*str, optional*) – The border noise removal method to be applied if *removeS1BorderNoise* is True. See *pyroSAR.S1.removeGRDBorderNoise()* for details. One of the following:
  - 'ESA': the pure implementation as described by ESA
  - 'pyroSAR': the ESA method plus the custom pyroSAR refinement (default)
- **removeS1ThermalNoise** (*bool, optional*) – Enables removal of S1 thermal noise (default).
- **offset** (*tuple, optional*) – A tuple defining offsets for left, right, top and bottom in pixels, e.g. (100, 100, 0, 0); this variable is overridden if a shapefile is defined. Default is None.
- **allow_RES_OSV** (*bool*) – (only applies to Sentinel-1) Also allow the less accurate RES orbit files to be used? The function first tries to download a POE file for the scene. If this fails and RES files are allowed, it will download the RES file. The selected OSV type is written to the workflow XML file. Processing is aborted if the correction fails (Apply-Orbit-File parameter continueOnFail set to false).
- **demName** (*str*) – The name of the auto-download DEM. Default is 'SRTM 1Sec HGT'. Is ignored when *externalDEMFile* is not None. Supported options:
  - ACE2_5Min
  - ACE30
  - ASTER 1sec GDEM
  - CDEM
  - Copernicus 30m Global DEM
  - Copernicus 90m Global DEM
  - GETASSE30
  - SRTM 1Sec Grid
  - SRTM 1Sec HGT
  - SRTM 3Sec
- **externalDEMFile** (*str or None, optional*) – The absolute path to an external DEM file. Default is None. Overrides *demName*.
- **externalDEMNoDataValue** (*int, float or None, optional*) – The no data value of the external DEM. If not specified (default) the function will try to read it from the specified external DEM.
- **externalDEMApplyEGM** (*bool, optional*) – Apply Earth Gravitational Model to external DEM? Default is True.
- **terrainFlattening** (*bool*) – Apply topographic normalization on the data?
- **basename_extensions** (*list of str or None*) – Names of additional parameters to append to the basename, e.g. ['orbitNumber_rel'].
- **test** (*bool, optional*) – If set to True the workflow xml file is only written and not executed. Default is False.

- **export_extra** (*list or None*) – A list of image file IDs to be exported to outdir. The following IDs are currently supported:
  - incidenceAngleFromEllipsoid
  - localIncidenceAngle
  - projectedLocalIncidenceAngle
  - DEM
  - layoverShadowMask
  - scatteringArea (requires `terrainFlattening=True`)
  - gammaSigmaRatio (requires `terrainFlattening=True` and `refarea=['sigma0', 'gamma0']`)
- **groupsize** (*int*) – The number of workers executed together in one gpt call.
- **cleanup** (*bool*) – Should all files written to the temporary directory during function execution be deleted after processing? Default is True.
- **tmpdir** (*str or None*) – Path of custom temporary directory, useful to separate output folder and temp folder. If *None*, the *outdir* location will be used. The created subdirectory will be deleted after processing if `cleanup=True`.
- **gpt_exceptions** (*dict or None*) – A dictionary to override the configured GPT executable for certain operators; each (sub-)workflow containing this operator will be executed with the define executable;
  - e.g. `{'Terrain-Flattening': '/home/user/snap/bin/gpt'}`
- **gpt_args** (*list or None*) – A list of additional arguments to be passed to the gpt call.
  - e.g. `['-x', '-c', '2048M']` for increased tile cache size and intermediate clearing
- **returnWF** (*bool*) – Return the full name of the written workflow XML file?
- **nodataValueAtSea** (*bool*) – Mask pixels acquired over sea? The sea mask depends on the selected DEM.
- **demResamplingMethod** (*str*) – One of the following:
  - 'NEAREST_NEIGHBOUR'
  - 'BILINEAR_INTERPOLATION'
  - 'CUBIC_CONVOLUTION'
  - 'BISINC_5_POINT_INTERPOLATION'
  - 'BISINC_11_POINT_INTERPOLATION'
  - 'BISINC_21_POINT_INTERPOLATION'
  - 'BICUBIC_INTERPOLATION'
- **imgResamplingMethod** (*str*) – The resampling method for geocoding the SAR image; the options are identical to demResamplingMethod.
- **alignToStandardGrid** (*bool*) – Align all processed images to a common grid?
- **standardGridOriginX** (*int or float*) – The x origin value for grid alignment
- **standardGridOriginY** (*int or float*) – The y origin value for grid alignment
- **speckleFilter** (*str*) – One of the following:
  - 'Boxcar'
  - 'Median'
  - 'Frost'

  - 'Gamma Map'
  - 'Refined Lee'
  - 'Lee'
  - 'Lee Sigma'
- **refarea** (*str or list*) – 'sigma0', 'gamma0' or a list of both
- **clean_edges** (*bool*) – erode noisy image edges? See *pyroSAR.snap.auxil.erode_edges()*. Does not apply to layover-shadow mask.
- **clean_edges_npixels** (*int*) – the number of pixels to erode.
- **rlks** (*int or None*) – the number of range looks. If not None, overrides the computation done by function *pyroSAR.ancillary.multilook_factors()* based on the image pixel spacing and the target spacing.
- **azlks** (*int or None*) – the number of azimuth looks. Like *rlks*.
- **dem_oversampling_multiple** (*int*) – a factor to multiply the DEM oversampling factor computed by SNAP. Used only for terrain flattening. The SNAP default of 1 has been found to be insufficient with stripe artifacts remaining in the image.
- **s1_osv_url_option** (*int*) – the OSV download URL option; see *pyroSAR.S1.OSV.catch()*

Returns
Either the name of the workflow file if `returnWF == True` or None otherwise

Return type
str or None

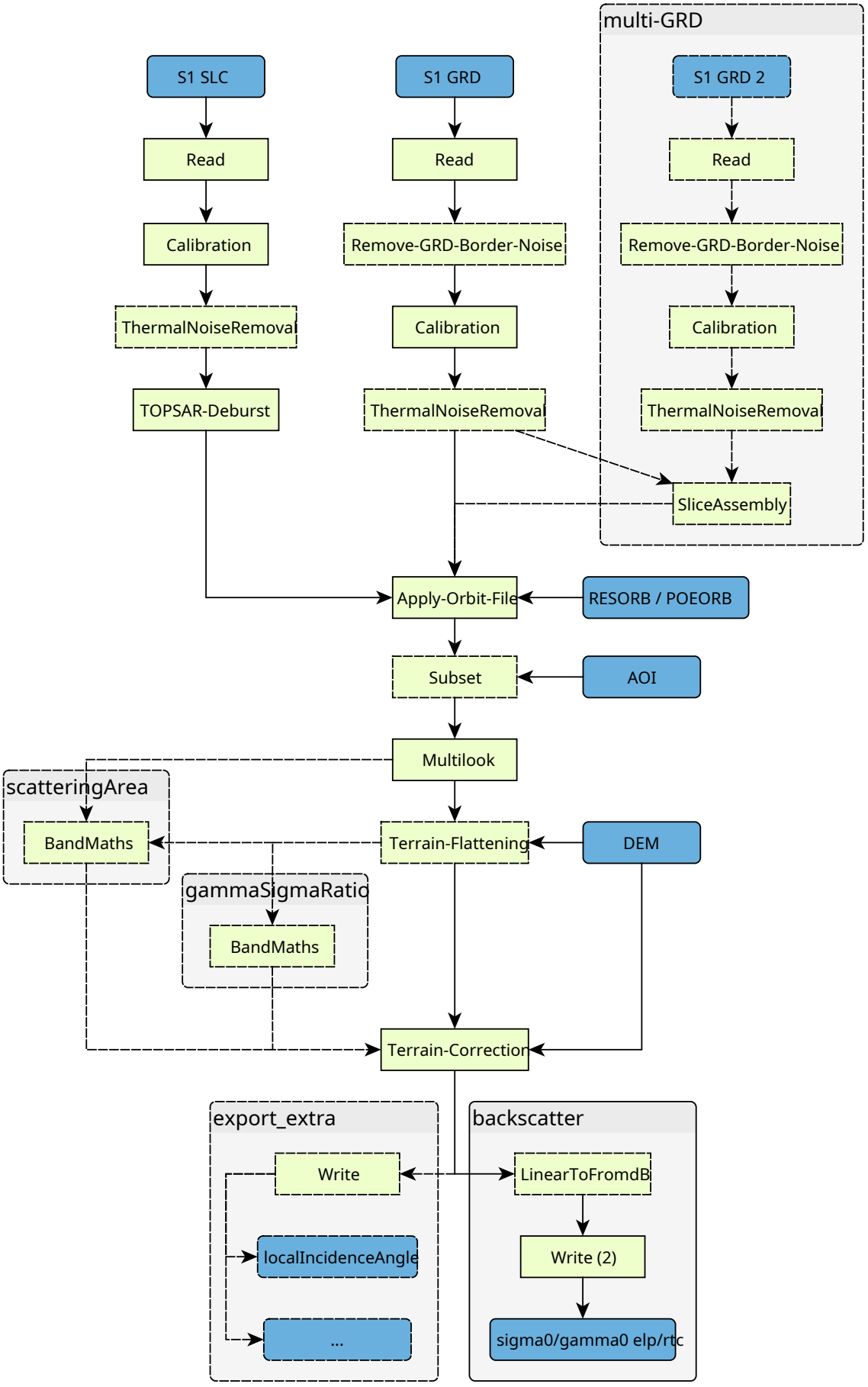

Fig. 1: Function geocode workflow diagram for processing Sentinel-1 scenes. Dashed lines depict optional steps. The output is sigma or gamma nought backscatter with ellipsoid or radiometric terrain correction (suffix elp/rtc) as well as several optional ancillary datasets (controlled via argument *export_extra*).

Examples

geocode a Sentinel-1 scene and export the local incidence angle map with it

```
>>> from pyroSAR.snap import geocode
>>> filename = 'S1A_IW_GRDH_1SDV_20180829T170656_20180829T170721_023464_028DE0_
↪F7BD.zip'
>>> geocode(infile=filename, outdir='outdir', spacing=20, scaling='dB',
>>>         export_extra=['DEM', 'localIncidenceAngle'], t_srs=4326)
```

See also:

pyroSAR.drivers.ID, spatialist.vector.Vector, spatialist.auxil.crsConvert()

pyroSAR.snap.util.**noise_power**(*infile*, *outdir*, *polarizations*, *spacing*, *t_srs*, *refarea='sigma0'*, *tmpdir=None*, *test=False*, *cleanup=True*, *demName='SRTM 1Sec HGT'*, *externalDEMFile=None*, *externalDEMNoDataValue=None*, *externalDEMApplyEGM=True*, *alignToStandardGrid=False*, *standardGridOriginX=0*, *standardGridOriginY=0*, *groupsize=1*, *clean_edges=False*, *clean_edges_npixels=1*, *rlks=None*, *azlks=None*, *osv_url_option=1*)

Generate Sentinel-1 noise power images for each polarization, calibrated to either beta, sigma or gamma nought. The written GeoTIFF files will carry the suffix NEBZ, NESZ or NEGZ respectively.

Parameters

- **infile** (*str*) – The SAR scene(s) to be processed
- **outdir** (*str*) – The directory to write the final files to.
- **polarizations** (*list[str]*) – The polarizations to be processed, e.g. ['VV', 'VH'].
- **spacing** (*int or float*) – The target pixel spacing in meters.
- **t_srs** (*int or str or osgeo.osr.SpatialReference*) – A target spatial reference system in WKT, EPSG, PROJ4 or OPENGIS format.
- **refarea** (*str*) – either 'beta0', 'gamma0' or 'sigma0'.
- **tmpdir** (*str*) – Path of custom temporary directory, useful to separate output folder and temp folder. If *None*, the *outdir* location will be used. The created subdirectory will be deleted after processing if `cleanup=True`.
- **test** (*bool*) – If set to True the workflow xml file is only written and not executed. Default is False.
- **cleanup** (*bool*) – Should all files written to the temporary directory during function execution be deleted after processing? Default is True.
- **demName** (*str*) – The name of the auto-download DEM. Default is 'SRTM 1Sec HGT'. Is ignored when *externalDEMFile* is not None. Supported options:
  - ACE2_5Min
  - ACE30
  - ASTER 1sec GDEM
  - CDEM
  - Copernicus 30m Global DEM
  - Copernicus 90m Global DEM
  - GETASSE30
  - SRTM 1Sec Grid
  - SRTM 1Sec HGT

– SRTM 3Sec

- **externalDEMFile** (*str or None, optional*) – The absolute path to an external DEM file. Default is None. Overrides *demName*.
- **externalDEMNoDataValue** (*int, float or None, optional*) – The no data value of the external DEM. If not specified (default) the function will try to read it from the specified external DEM.
- **externalDEMApplyEGM** (*bool, optional*) – Apply Earth Gravitational Model to external DEM? Default is True.
- **alignToStandardGrid** (*bool*) – Align all processed images to a common grid?
- **standardGridOriginX** (*int or float*) – The x origin value for grid alignment
- **standardGridOriginY** (*int or float*) – The y origin value for grid alignment
- **groupsize** (*int*) – The number of workers executed together in one gpt call.
- **clean_edges** (*bool*) – erode noisy image edges? See *pyroSAR.snap.auxil.erode_edges()*. Does not apply to layover-shadow mask.
- **clean_edges_npixels** (*int*) – the number of pixels to erode.
- **rlks** (*int or None*) – the number of range looks. If not None, overrides the computation done by function *pyroSAR.ancillary.multilook_factors()* based on the image pixel spacing and the target spacing.
- **azlks** (*int or None*) – the number of azimuth looks. Like *rlks*.
- **osv_url_option** (*int*) – the OSV download URL option; see *pyroSAR.S1.OSV.catch()*

2.2.2 Workflow Parsing and Execution

gpt	Wrapper for ESA SNAP's Graph Processing Tool GPT.
execute	execute SNAP workflows via the Graph Processing Tool GPT.
parse_node	parse an XML node recipe.
parse_recipe	parse a SNAP recipe
split	split a workflow file into groups and write them to separate workflows including source and write target linking.
groupbyWorkers	split a SNAP workflow into groups containing a maximum defined number of operators.
Workflow	Class for convenient handling of SNAP XML workflows
Node	class for handling of SNAP workflow processing nodes
Par	class for handling processing node parameters
Par_BandMath	class for handling BandMaths node parameters
dem_parametrize	DEM parametrization for a full workflow or a single node.
geo_parametrize	convenience function for parametrizing geocoding nodes.
mli_parametrize	Convenience function for parametrizing a *Multilook* node.
orb_parametrize	convenience function for parametrizing an *Apply-Orbit-File*.
sub_parametrize	convenience function for parametrizing an *Subset* node.

class pyroSAR.snap.auxil.**Node**(*element*)

Bases: object

class for handling of SNAP workflow processing nodes

Parameters
element (*Element*) – the node XML element

copy()

Returns
a copy of the Node object

Return type
Node

property id

Returns
the node ID

Return type
str

property operator

Returns
the name of the node's processing operator

Return type
str

property parameters

Returns
the processing parameters of the node

Return type
Par or *Par_BandMath*

property source

Returns
the ID(s) of the source node(s)

Return type
str or list

class pyroSAR.snap.auxil.**Par**(*operator*, *element*)

Bases: object

class for handling processing node parameters

Parameters

- **operator** (*str*) – the name of the SNAP Node operator
- **element** (*Element*) – the node parameter XML element

__getitem__(*item*)

Parameters
item –

Return type
str

dict()

Returns
the parameters as a dictionary

Return type
dict

items()

Returns
the parameters as (key, value) as from `dict.items()`

Return type
list

keys()

Returns
the parameter names as from `dict.keys()`

Return type
list

values()

Returns
the parameter values as from `dict.values()`

Return type
list

class pyroSAR.snap.auxil.**Par_BandMath**(*operator*, *element*)

Bases: *Par*

class for handling BandMaths node parameters

Parameters
element (*Element*) – the node parameter XML element

add_equation()

add an equation element to the node

clear_variables()

remove all *variables* elements from the node

class pyroSAR.snap.auxil.**Workflow**(*xmlfile*)

Bases: `object`

Class for convenient handling of SNAP XML workflows

Parameters
xmlfile (*str*) – the workflow XML file

property ids

Returns
the IDs of all nodes

Return type
list

index(*node*)

Parameters
node (Node) – a node in the workflow

Returns
the index position of the node in the workflow

Return type
int

insert_node(*node*, *before=None*, *after=None*, *resetSuccessorSource=True*, *void=True*)

insert one or multiple node(s) into the workflow including setting the source to the predecessor and setting the ID as source of the successor.

Parameters

- **node** (*Node or list[Node]*) – the node(s) to be inserted
- **before** (*Node, str or list*) – a Node object; the ID(s) of the node(s) before the newly inserted node; a list of node IDs is intended for nodes that require multiple sources, e.g. sliceAssembly
- **after** (*Node, str*) – a Node object; the ID of the node after the newly inserted node
- **resetSuccessorSource** (*bool*) – reset the source of the successor node to the ID of the newly inserted node?
- **void** (*bool*) – if false, the function returns the node

Returns
the new node, a list of nodes, or None, depending on the *node* input and argument *void*

Return type
Node or list[*Node*] or None

nodes()

Returns
the list of *Node* objects in the workflow

Return type
list[*Node*]

property operators

Returns
the names of the unique operators in the workflow

Return type
list

refresh_ids()

Ensure unique IDs for all nodes. If two nodes with the same ID are found one is renamed to "ID (2)". E.g. 2 x "Write" -> "Write", "Write (2)". This method is no longer used and is just kept in case there is need for it in the future.

set_par(*key*, *value*, *exceptions=None*)

set a parameter for all nodes in the workflow

Parameters

- **key** (*str*) – the parameter name
- **value** (*bool or int or float or str*) – the parameter value
- **exceptions** (*list*) – a list of node IDs whose parameters should not be changed

successors(*id*, *recursive=False*)

find the succeeding node(s) of a node

Parameters

- **id** (*str*) – the ID of the node

- **recursive** (*bool*) – find successors recursively?

Returns
the ID(s) of the successors

Return type
list of str

suffix(*stop=None*)

Get the SNAP operator suffix sequence

Parameters
stop (*str*) – the ID of the last workflow node

Returns
a file suffix created from the order of which the nodes will be executed

Return type
str

write(*outfile*)

write the workflow to an XML file

Parameters
outfile (*str*) – the name of the file to write

pyroSAR.snap.auxil.**dem_parametrize**(*workflow=None*, *node=None*, *demName='SRTM 1Sec HGT'*, *externalDEMFile=None*, *externalDEMNoDataValue=None*, *externalDEMApplyEGM=False*, *demResamplingMethod='BILINEAR_INTERPOLATION'*)

DEM parametrization for a full workflow or a single node. In the former case, all nodes with the DEM-relevant parameters can be modified at once, e.g. *Terrain-Flattening* and *Terrain-Correction*.

Parameters

- **workflow** (*Workflow or None*) – a SNAP workflow object
- **node** (*Node or None*) – a SNAP node object
- **demName** (*str*) – The name of the auto-download DEM. Default is 'SRTM 1Sec HGT'. Is ignored when *externalDEMFile* is not None. Supported options:
  - ACE2_5Min
  - ACE30
  - ASTER 1sec GDEM
  - CDEM
  - Copernicus 30m Global DEM
  - Copernicus 90m Global DEM
  - GETASSE30
  - SRTM 1Sec Grid
  - SRTM 1Sec HGT
  - SRTM 3Sec
- **externalDEMFile** (*str or None, optional*) – The absolute path to an external DEM file. Default is None. Overrides *demName*.
- **externalDEMNoDataValue** (*int, float or None, optional*) – The no data value of the external DEM. If not specified (default) the function will try to read it from the specified external DEM.

- **externalDEMApplyEGM** (*bool, optional*) – Apply Earth Gravitational Model to external DEM? Default is True.
- **demResamplingMethod** (*str*) – the DEM resampling method

pyroSAR.snap.auxil.**execute**(*xmlfile*, *cleanup=True*, *gpt_exceptions=None*, *gpt_args=None*)

execute SNAP workflows via the Graph Processing Tool GPT. This function merely calls gpt with some additional command line arguments and raises a RuntimeError on fail. This function is used internally by function *gpt()*.

Parameters

- **xmlfile** (*str*) – the name of the workflow XML file
- **cleanup** (*bool*) – should all files written to the temporary directory during function execution be deleted after processing?
- **gpt_exceptions** (*dict*) – a dictionary to override the configured GPT executable for certain operators; each (sub-)workflow containing this operator will be executed with the define executable;
  - e.g. `{'Terrain-Flattening': '/home/user/snap/bin/gpt'}`
- **gpt_args** (*list or None*) – a list of additional arguments to be passed to the GPT call
  - e.g. `['-x', '-c', '2048M']` for increased tile cache size and intermediate clearing

Raises

RuntimeError –

pyroSAR.snap.auxil.**geo_parametrize**(*spacing*, *t_srs*, *tc_method='Range-Doppler'*, *sourceBands=None*, *demName='SRTM 1Sec HGT'*, *externalDEMFile=None*, *externalDEMNoDataValue=None*, *externalDEMApplyEGM=True*, *alignToStandardGrid=False*, *standardGridAreaOrPoint='point'*, *standardGridOriginX=0*, *standardGridOriginY=0*, *nodataValueAtSea=False*, *export_extra=None*, *demResamplingMethod='BILINEAR_INTERPOLATION'*, *imgResamplingMethod='BILINEAR_INTERPOLATION'*, ***kwargs*)

convenience function for parametrizing geocoding nodes.

Parameters

- **workflow** (Workflow) – the SNAP workflow object
- **before** (*str*) – the ID of the node after which the terrain correction node will be inserted
- **tc_method** (*str*) – the terrain correction method. Supported options:
  - Range-Doppler (SNAP node *Terrain-Correction*)
  - SAR simulation cross correlation (SNAP nodes *SAR-Simulation*->`Cross-Correlation`->`SARSim-Terrain-Correction`)
- **sourceBands** (*list[str] or None*) – the image band names to geocode; default None: geocode all incoming bands.
- **spacing** (*int or float*) – The target pixel spacing in meters.
- **t_srs** (*int or str or osgeo.osr.SpatialReference*) – A target geographic reference system in WKT, EPSG, PROJ4 or OPENGIS format. See function `spatialist.auxil.crsConvert()` for details.
- **demName** (*str*) – The name of the auto-download DEM. Default is 'SRTM 1Sec HGT'. Is ignored when *externalDEMFile* is not None. Supported options:
  - ACE2_5Min

  - ACE30
  - ASTER 1sec GDEM
  - CDEM
  - Copernicus 30m Global DEM
  - Copernicus 90m Global DEM
  - GETASSE30
  - SRTM 1Sec Grid
  - SRTM 1Sec HGT
  - SRTM 3Sec
- **externalDEMFile** (*str or None, optional*) – The absolute path to an external DEM file. Default is None. Overrides *demName*.
- **externalDEMNoDataValue** (*int, float or None, optional*) – The no data value of the external DEM. If not specified (default) the function will try to read it from the specified external DEM.
- **externalDEMApplyEGM** (*bool, optional*) – Apply Earth Gravitational Model to external DEM? Default is True.
- **alignToStandardGrid** (*bool*) – Align all processed images to a common grid?
- **standardGridAreaOrPoint** (*str*) – treat alignment coordinate as pixel center ('point', SNAP default) or upper left ('area').
- **standardGridOriginX** (*int or float*) – The x origin value for grid alignment
- **standardGridOriginY** (*int or float*) – The y origin value for grid alignment
- **nodataValueAtSea** (*bool*) – mask values over sea?
- **export_extra** (*list[str] or None*) – a list of ancillary layers to write. Supported options:
  - DEM
  - latLon
  - incidenceAngleFromEllipsoid (Range-Doppler only)
  - layoverShadowMask
  - localIncidenceAngle
  - projectedLocalIncidenceAngle
  - selectedSourceBand
- **demResamplingMethod** (*str*) – the DEM resampling method
- **imgResamplingMethod** (*str*) – the image resampling method
- **kwargs** – further keyword arguments for node parametrization. Known options:
  - outputComplex
  - applyRadiometricNormalization
  - saveSigmaNought
  - saveGammaNought
  - saveBetaNought
  - incidenceAngleForSigma0
  - incidenceAngleForGamma0

- auxFile
- externalAuxFile
- openShiftsFile (SAR simulation cross correlation only)
- openResidualsFile (SAR simulation cross correlation only)

Returns
the Terrain-Correction node object or a list containing the objects for SAR-Simulation, Cross-Correlation and SARSim-Terrain-Correction.

Return type
Node or list[*Node*]

pyroSAR.snap.auxil.**gpt**(*xmlfile*, *tmpdir*, *groups=None*, *cleanup=True*, *gpt_exceptions=None*, *gpt_args=None*, *removeS1BorderNoiseMethod='pyroSAR'*)

Wrapper for ESA SNAP's Graph Processing Tool GPT. Input is a readily formatted workflow XML file as for example created by function *geocode()*. Additional to calling GPT, this function will

- (if processing Sentinel-1 GRD data with IPF version <2.9 and `removeS1BorderNoiseMethod='pyroSAR'`) unpack the scene and perform the custom removal (*pyroSAR.S1.removeGRDBorderNoise()*).
- if *groups* is not None:
  - split the workflow into sub-workflows (*pyroSAR.snap.auxil.split()*)
  - execute the sub-workflows (*pyroSAR.snap.auxil.execute()*)

Note: Depending on the parametrization this function might create two subdirectories in *tmpdir*, bnr for S1 GRD border noise removal and sub for sub-workflows and their intermediate outputs. Both are deleted if `cleanup=True`. If *tmpdir* is empty afterward it is also deleted.

Parameters

- **xmlfile** (*str*) – the name of the workflow XML file
- **tmpdir** (*str*) – a temporary directory for storing intermediate files
- **groups** (*list[list[str]] or None*) – a list of lists each containing IDs for individual nodes. If not None, the workflow is split into sub-workflows executing the nodes in the respective group. These workflows and their output products are stored into the subdirectory sub of *tmpdir*.
- **cleanup** (*bool*) – should all temporary files be deleted after processing? First, the subdirectories bnr and sub of *tmpdir* are deleted. If *tmpdir* is empty afterward it is also deleted.
- **gpt_exceptions** (*dict or None*) – a dictionary to override the configured GPT executable for certain operators; each (sub-)workflow containing this operator will be executed with the define executable;
  - e.g. `{'Terrain-Flattening': '/home/user/snap/bin/gpt'}`
- **gpt_args** (*list[str] or None*) – a list of additional arguments to be passed to the gpt call
  - e.g. `['-x', '-c', '2048M']` for increased tile cache size and intermediate clearing
- **removeS1BorderNoiseMethod** (*str*) – the border noise removal method to be applied, See *pyroSAR.S1.removeGRDBorderNoise()* for details; one of the following:
  - 'ESA': the pure implementation as described by ESA

- 'pyroSAR': the ESA method plus the custom pyroSAR refinement. This is only applied if the IPF version is < 2.9 where additional noise removal was necessary. The output of the additional noise removal is stored in the subdirectory bnr of *tmpdir*.

pyroSAR.snap.auxil.**groupbyWorkers**(*xmlfile*, *n=2*)

split a SNAP workflow into groups containing a maximum defined number of operators.

Parameters

- **xmlfile** (*str*) – the SNAP xml workflow
- **n** (*int*) – the maximum number of worker nodes in each group; Read, Write and BandSelect are excluded.

Returns

a list of lists each containing the IDs of all nodes belonging to the groups including Read and Write nodes; this list can e.g. be passed to function *split()* to split the workflow into new sub-workflow files based on the newly created groups or directly to function *gpt()*, which will call *split()* internally.

Return type

list[list[str]]

pyroSAR.snap.auxil.**mli_parametrize**(*scene*, *spacing=None*, *rlks=None*, *azlks=None*, ***kwargs*)

Convenience function for parametrizing a *Multilook* node.

Parameters

- **scene** (pyroSAR.drivers.ID) – The SAR scene to be processed
- **spacing** (*int or float or None*) – the target pixel spacing for automatic determination of looks using function *multilook_factors()*. Overridden by arguments *rlks* and *azlks* if they are not None.
- **rlks** (*int or None*) – the number of range looks
- **azlks** (*int or None*) – the number of azimuth looks
- **bands** (*list[str] or None*) – an optional list of bands names
- **kwargs** – further keyword arguments for node parametrization. Known options:
  - grSquarePixel
  - outputIntensity
  - sourceBands

Returns

either a *Node* object if multilooking is necessary (either *rlks* or *azlks* are greater than 1) or None.

Return type

Node or None

See also:

pyroSAR.ancillary.multilook_factors

pyroSAR.snap.auxil.**orb_parametrize**(*scene*, *formatName*, *allow_RES_OSV=True*, *url_option=1*, ***kwargs*)

convenience function for parametrizing an *Apply-Orbit-File*. Required Sentinel-1 orbit files are directly downloaded.

Parameters

- **scene** (pyroSAR.drivers.ID) – The SAR scene to be processed
- **workflow** (Workflow) – the SNAP workflow object

- **before** (*str*) – the ID of the node after which the *Apply-Orbit-File* node will be inserted
- **formatName** (*str*) – the scene's data format
- **allow_RES_OSV** (*bool*) – (only applies to Sentinel-1) Also allow the less accurate RES orbit files to be used?
- **url_option** (*int*) – the OSV download URL option; see *pyroSAR.S1.OSV.catch()*
- **kwargs** – further keyword arguments for node parametrization. Known options:
  - continueOnFail
  - polyDegree

Returns
the Apply-Orbit-File node object

Return type
Node

pyroSAR.snap.auxil.**parse_node**(*name*, *use_existing=True*)

parse an XML node recipe. The XML representation and parameter default values are read from the docstring of an individual node by calling *gpt <node> -h*. The result is then written to an XML text file under *$HOME/.pyroSAR/snap/nodes* which is subsequently read for parsing instead of again calling *gpt*.

Parameters

- **name** (*str*) – the name of the processing node, e.g. Terrain-Correction
- **use_existing** (*bool*) – use an existing XML text file or force re-parsing the gpt docstring and overwriting the XML file?

Returns
the parsed node

Return type
Node

Examples

```
>>> tnr = parse_node('ThermalNoiseRemoval')
>>> print(tnr.parameters)
{'selectedPolarisations': None, 'removeThermalNoise': 'true',
↪'reIntroduceThermalNoise': 'false'}
```

pyroSAR.snap.auxil.**parse_recipe**(*name*)

parse a SNAP recipe

Parameters
name (*str*) –

the name of the recipe; current options:

- *blank*: a workflow without any nodes
- *geocode*: a basic workflow containing *Read*, *Apply-Orbit-File*, *Calibration*, *Terrain-Flattening* and *Write* nodes

Returns
the parsed recipe

Return type
Workflow

Examples

```
>>> from pyroSAR.snap.auxil import parse_recipe
>>> workflow = parse_recipe('base')
```

pyroSAR.snap.auxil.**split**(*xmlfile*, *groups*, *outdir=None*)

split a workflow file into groups and write them to separate workflows including source and write target linking. The new workflows are written to a sub-directory *temp* of the target directory defined in the input's *Write* node. Each new workflow is parameterized with a *Read* and *Write* node if they don't already exist. Temporary outputs are written to *BEAM-DIMAP* files named after the workflow suffix sequence.

Parameters

- **xmlfile** (*str*) – the workflow to be split
- **groups** (*list*) – a list of lists each containing IDs for individual nodes
- **outdir** (*str or None*) – the directory into which to write the XML workflows and the intermediate files created by them. If None, the name will be created from the file name of the node with ID 'Write', which is treated as a directory, and a subdirectory 'tmp'.

Returns
the names of the newly written temporary workflows

Return type
list of str

Raises
RuntimeError –

pyroSAR.snap.auxil.**sub_parametrize**(*scene*, *geometry=None*, *offset=None*, *buffer=0.01*, *copyMetadata=True*, ***kwargs*)

convenience function for parametrizing an *Subset* node.

Parameters

- **scene** (pyroSAR.drivers.ID) – The SAR scene to be processed
- **geometry** (*dict or spatialist.vector.Vector or str or None*) – A vector geometry for geographic subsetting (node parameter geoRegion):
  - Vector: a vector object in arbitrary CRS
  - str: a name of a file that can be read with Vector in arbitrary CRS
  - dict: a dictionary with keys *xmin*, *xmax*, *ymin*, *ymax* in EPSG:4326 coordinates
- **offset** (*tuple or None*) – a tuple with pixel coordinates as (left, right, top, bottom)
- **buffer** (*int or float*) – an additional buffer in degrees to add around the *geometry*
- **copyMetadata** (*bool*) – copy the metadata of the source product?
- **kwargs** – further keyword arguments for node parametrization. Known options:
  - fullSwath
  - referenceBand
  - sourceBands
  - subSamplingX
  - subSamplingY
  - tiePointGrids

Returns
the Subset node object

Return type
Node

2.2.3 General Utilities

erode_edges	Erode noisy edge pixels in SNAP-processed images.
writer	SNAP product writing utility

pyroSAR.snap.auxil.**erode_edges**(*src*, *only_boundary=False*, *connectedness=4*, *pixels=1*)

Erode noisy edge pixels in SNAP-processed images. It was discovered that images contain border pixel artifacts after *Terrain-Correction*. Likely this is coming from treating the value 0 as regular value instead of no data during resampling. This function erodes these edge pixels using `scipy.ndimage.binary_erosion()`. scipy is not a base dependency of pyroSAR and has to be installed separately.

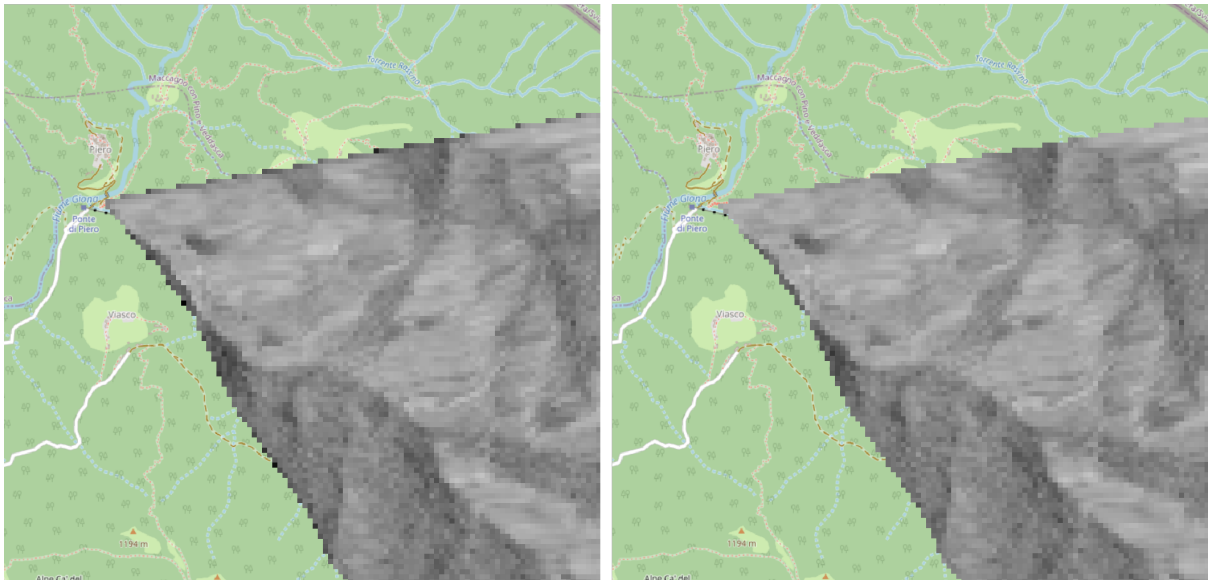

Fig. 2: VV gamma0 RTC backscatter image visualizing the noisy border (left) and the cleaned result (right). The area covers approx. 2.3 x 2.3 km^2. Pixel spacing is 20 m. connectedness 4, 1 pixel.

Parameters

- **src** (*str*) – a processed SAR image in BEAM-DIMAP format (.dim), a single .img file (ENVI format) or a directory with .img files. 0 is assumed as no data value.
- **only_boundary** (*bool*) – only erode edges at the image boundary (or also at data gaps caused by e.g. masking during Terrain-Flattening)?
- **connectedness** (*int*) – the number of pixel neighbors considered for the erosion. Either 4 or 8, translating to a `scipy.ndimage.generate_binary_structure()` *connectivity* of 1 or 2, respectively.
- **pixels** (*int*) – the number of pixels to erode from the edges. Directly translates to *iterations* of `scipy.ndimage.iterate_structure()`.

pyroSAR.snap.auxil.**writer**(*xmlfile*, *outdir*, *basename_extensions=None*, *clean_edges=False*, *clean_edges_npixels=1*)

SNAP product writing utility

Parameters

- **xmlfile** (*str*) – the name of the workflow XML file.

- **outdir** (*str*) – the directory into which to write the final files.
- **basename_extensions** (*list of str or None*) – names of additional parameters to append to the basename, e.g. `['orbitNumber_rel']`.
- **clean_edges** (*bool*) – erode noisy image edges? See *pyroSAR.snap.auxil.erode_edges()*. Does not apply to layover-shadow mask.
- **clean_edges_npixels** (*int*) – the number of pixels to erode.

2.3 GAMMA

2.3.1 Processing

calibrate	radiometric calibration of SAR scenes
convert2gamma	general function for converting SAR images to GAMMA format
correctOSV	correct GAMMA parameter files with orbit state vector information from dedicated OSV files; OSV files are downloaded automatically to either the defined *osvdir* or relative to the user's home directory: *~/.snap/auxdata/Orbits/Sentinel-1*.
geocode	general function for radiometric terrain correction (RTC) and geocoding of SAR backscatter images with GAMMA.
ISPPar	Reader for ISP parameter files of the GAMMA software package
multilook	Multilooking of SLC and MLI images.
ovs	compute DEM oversampling factors for a target resolution in meters
par2hdr	Create an ENVI HDR file from a GAMMA PAR file
process	wrapper function to execute GAMMA commands via module `subprocess`
S1_deburst	Debursting of Sentinel-1 SLC imagery in GAMMA
UTM	convert a gamma parameter file corner coordinate from EQA to UTM

class pyroSAR.gamma.**ISPPar**(*filename*)

Bases: `object`

Reader for ISP parameter files of the GAMMA software package

This class allows to read all information from files in GAMMA's parameter file format. Each key-value pair is parsed and added as attribute. For instance if the parameter file contains the pair 'sensor: TSX-1' an attribute named 'sensor' with the value 'TSX-1' will be available.

The values are converted to native Python types, while unit identifiers like 'dB' or 'Hz' are removed. Please see the GAMMA reference manual for further information on the actual file format.

Parameters

filename (*str*) – the GAMMA parameter file

Examples

```
>>> from pyroSAR.gamma import ISPPar
>>> with ISPPar('S1A__IW___A_20141115T181801_VH_grd.par') as par:
...     print(par) # print an overview of all available metadata
...     print(par.keys) # print all parameter names
...     for key, value in par.envidict().items():
...         print('{0}: {1}'.format(key, value)) # print the ENVI HDR compliant␣
↪metadata
```

keys

the names of all parameters

Type
list

envidict(*nodata=None*)

export relevant metadata to an ENVI HDR file compliant format

Parameters
nodata (*int, float or None*) – a no data value to write to the HDR file via attribute 'data ignore value'

Returns
a dictionary containing attributes translated to ENVI HDR naming

Return type
dict

pyroSAR.gamma.**S1_deburst**(*burst1*, *burst2*, *burst3*, *name_out*, *rlks=5*, *azlks=1*, *replace=False*, *logpath=None*, *outdir=None*, *shellscript=None*)

Debursting of Sentinel-1 SLC imagery in GAMMA

The procedure consists of two steps. First antenna pattern deramping and then mosaicing of the single deramped bursts. For mosaicing, the burst boundaries are calculated from the number of looks in range (*rlks*) and azimuth (*azlks*), in this case 5 range looks and 1 azimuth looks. Alternately 10 range looks and 2 azimuth looks could be used.

Parameters

- **burst1** (*str*) – burst image 1
- **burst2** (*str*) – burst image 2
- **burst3** (*str*) – burst image 3
- **name_out** (*str*) – the name of the output file
- **rlks** (*int*) – the number of looks in range
- **azlks** (*int*) – the number of looks in azimuth
- **replace** (*bool*) – replace the burst images by the new file? If True, the three burst images will be deleted.
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

class pyroSAR.gamma.**UTM**(*parfile*)

Bases: object

convert a gamma parameter file corner coordinate from EQA to UTM

Parameters
parfile (*str*) – the GAMMA parameter file to read the coordinate from

Example

```
>>> from pyroSAR.gamma import UTM
>>> print(UTM('gamma.par').zone)
```

pyroSAR.gamma.**calibrate**(*id*, *directory*, *return_fnames=False*, *logpath=None*, *outdir=None*, *shellscript=None*)

radiometric calibration of SAR scenes

Parameters

- **id** (*ID*) – an SAR scene object of type pyroSAR.ID or any subclass
- **directory** (*str*) – the directory to search for GAMMA calibration candidates
- **return_fnames** (*bool*) – return the names of the output image files? Default: False.
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the GAMMA commands to in shell format

pyroSAR.gamma.**convert2gamma**(*id*, *directory*, *S1_tnr=True*, *S1_bnr=True*, *basename_extensions=None*, *exist_ok=False*, *return_fnames=False*, *logpath=None*, *outdir=None*, *shellscript=None*)

general function for converting SAR images to GAMMA format

Parameters

- **id** (*ID*) – an SAR scene object of type pyroSAR.ID or any subclass
- **directory** (*str*) – the output directory for the converted images
- **S1_tnr** (*bool*) – only Sentinel-1: should thermal noise removal be applied to the image?
- **S1_bnr** (*bool*) – only Sentinel-1 GRD: should border noise removal be applied to the image? This is available since version 20191203, for older versions this argument is ignored.
- **basename_extensions** (*list[str] or None*) – names of additional parameters to append to the basename, e.g. ['orbitNumber_rel']
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?
- **return_fnames** (*bool*) – return the names of the output image files? Default: False.
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the GAMMA commands to in bash format

Returns
the sorted image file names if `return_fnames=True` and None otherwise

Return type
list[str] or None

pyroSAR.gamma.**correctOSV**(*id*, *directory*, *osvdir=None*, *osvType='POE'*, *timeout=20*, *logpath=None*, *outdir=None*, *shellscript=None*, *url_option=1*)

correct GAMMA parameter files with orbit state vector information from dedicated OSV files; OSV files are downloaded automatically to either the defined *osvdir* or relative to the user's home directory: *~/.snap/auxdata/Orbits/Sentinel-1*.

Parameters

- **id** (ID) – the scene to be corrected
- **directory** (*str or None*) – a directory to be scanned for files associated with the scene, e.g. an SLC in GAMMA format. If the OSV file is packed in a zip file it will be unpacked to a subdirectory *osv*.
- **osvdir** (*str*) – the directory of OSV files; subdirectories POEORB and RESORB are created automatically
- **osvType** (*{'POE', 'RES'}*) – the OSV type to be used
- **timeout** (*int or tuple or None*) – the timeout in seconds for downloading OSV files as provided to `requests.get()`
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the GAMMA commands to in shell format
- **url_option** (*int*) – the OSV download URL option; see *pyroSAR.S1.OSV.catch()*

Examples

```
>>> from pyroSAR import identify
>>> from pyroSAR.gamma import correctOSV, convert2gamma
>>> filename = 'S1A_IW_GRDH_1SDV_20150222T170750_20150222T170815_004739_005DD8_
↪3768.zip'
# identify the SAR scene
>>> scene = identify(filename)
# unpack the zipped scene to an arbitrary directory
>>> scene.unpack('/home/test')
>>> print(scene.scene)
/home/test/S1A_IW_GRDH_1SDV_20150222T170750_20150222T170815_004739_005DD8_3768.
↪SAFE
# convert the unpacked scene to GAMMA format
>>> convert2gamma(id=scene, directory=scene.scene)
# correct the OSV information of the converted GAMMA images
>>> correctOSV(id=scene, osvdir='/home/test/osv')
```

See also:

pyroSAR.drivers.SAFE.getOSV(), *pyroSAR.S1.OSV*

pyroSAR.gamma.**geocode**(*scene*, *dem*, *tmpdir*, *outdir*, *spacing*, *scaling='linear'*, *func_geoback=1*, *nodata=(0, -99)*, *update_osv=True*, *osvdir=None*, *allow_RES_OSV=False*, *cleanup=True*, *export_extra=None*, *basename_extensions=None*, *removeS1BorderNoiseMethod='gamma'*, *refine_lut=False*, *rlks=None*, *azlks=None*, *s1_osv_url_option=1*)

general function for radiometric terrain correction (RTC) and geocoding of SAR backscatter images with GAMMA. Applies the RTC method by Small [3] to retrieve gamma nought RTC backscatter.

Parameters

- **scene** (*str or ID or list*) – the SAR scene(s) to be processed
- **dem** (*str*) – the reference DEM in GAMMA format
- **tmpdir** (*str*) – a temporary directory for writing intermediate files
- **outdir** (*str*) – the directory for the final GeoTIFF output files
- **spacing** (*float or int*) – the target pixel spacing in meters
- **scaling** (*str or list[str]*) – the value scaling of the backscatter values; either 'linear', 'db' or a list of both, i.e. ['linear', 'db']
- **func_geoback** (*{0, 1, 2, 3, 4, 5, 6, 7}*) – backward geocoding interpolation mode (see GAMMA command *geocode_back*)
  - 0: nearest-neighbor
  - 1: bicubic spline (default)
  - 2: bicubic-spline, interpolate log(data)
  - 3: bicubic-spline, interpolate sqrt(data)
  - 4: B-spline interpolation (default B-spline degree: 5)
  - 5: B-spline interpolation sqrt(x) (default B-spline degree: 5)
  - 6: Lanczos interpolation (default Lanczos function order: 5)
  - 7: Lanczos interpolation sqrt(x) (default Lanczos function order: 5)

  Note: log and sqrt interpolation modes should only be used with non-negative data!

  Note: GAMMA recommendation for MLI data: "The interpolation should be performed on the square root of the data. A mid-order (3 to 5) B-spline interpolation is recommended."
- **nodata** (*tuple[float or int]*) – the nodata values for the output files; defined as a tuple with two values, the first for linear, the second for logarithmic scaling
- **update_osv** (*bool*) – update the orbit state vectors?
- **osvdir** (*str or None*) – a directory for Orbit State Vector files; this is currently only used by for Sentinel-1 where two subdirectories POEORB and RESORB are created; if set to None, a subdirectory OSV is created in the directory of the unpacked scene.
- **allow_RES_OSV** (*bool*) – also allow the less accurate RES orbit files to be used? Otherwise the function will raise an error if no POE file exists.
- **cleanup** (*bool*) – should all files written to the temporary directory during function execution be deleted after processing?
- **export_extra** (*list[str] or None*) – a list of image file IDs to be exported to outdir
  - format is GeoTIFF if the file is geocoded and ENVI otherwise. Non-geocoded images can be converted via GAMMA command data2tiff yet the output was found impossible to read with GIS software
  - scaling of SAR image products is applied as defined by parameter *scaling*
  - see Notes for ID options
- **basename_extensions** (*list[str] or None*) – names of additional parameters to append to the basename, e.g. ['orbitNumber_rel']

- **removeS1BorderNoiseMethod** (*str or None*) – the S1 GRD border noise removal method to be applied, See *pyroSAR.S1.removeGRDBorderNoise()* for details; one of the following:
  - 'ESA': the pure implementation as described by ESA
  - 'pyroSAR': the ESA method plus the custom pyroSAR refinement
  - 'gamma': the GAMMA implementation of [1]
  - None: do not remove border noise
- **refine_lut** (*bool*) – should the LUT for geocoding be refined using pixel area normalization?
- **rlks** (*int or None*) – the number of range looks. If not None, overrides the computation done by function *pyroSAR.ancillary.multilook_factors()* based on the image pixel spacing and the target spacing.
- **azlks** (*int or None*) – the number of azimuth looks. Like *rlks*.
- **s1_osv_url_option** (*int*) – the OSV download URL option; see *pyroSAR.S1.OSV.catch()*

Note:

intermediate output files
DEM products are named <scene identifier>_<ID>, e.g. *S1A__IW___A_20141012T162337_inc_geo*
SAR products will additionally contain the polarization, e.g.
S1A__IW___A_20141012T162337_VV_grd_mli
IDs in brackets are only written if selected by *export_extra*

- images in range-Doppler geometry
  - **grd**: the ground range detected SAR intensity image
  - **grd_mli**: the multi-looked grd image with approximated target resolution
  - (**pix_ellip_sigma0**): ellipsoid-based pixel area
  - (**pix_area_sigma0**): illuminated area as obtained from integrating DEM-facets in sigma projection (command pixel_area)
  - (**pix_area_gamma0**): illuminated area as obtained from integrating DEM-facets in gamma projection (command pixel_area)
  - **pix_ratio**: pixel area normalization factor (pix_ellip_sigma0 / pix_area_gamma0)
  - **grd_mli_gamma0-rtc**: the terrain-corrected gamma0 backscatter (grd_mli * pix_ratio)
  - (**gs_ratio**): gamma-sigma ratio (pix_gamma0 / pix_sigma0)
- images in map geometry
  - **dem_seg_geo**: dem subsetted to the extent of the intersection between input DEM and SAR image
  - (**u_geo**): zenith angle of surface normal vector n (angle between z and n)
  - (**v_geo**): orientation angle of n (between x and projection of n in xy plane)
  - **inc_geo**: local incidence angle (between surface normal and look vector)
  - (**psi_geo**): projection angle (between surface normal and image plane normal)
  - **ls_map_geo**: layover and shadow map
  - (**sim_sar_geo**): simulated SAR backscatter image
  - (**pix_ellip_sigma0_geo**): ellipsoid-based pixel area

  - (**pix_area_sigma0_geo**): illuminated area as obtained from integrating DEM-facets in sigma projection (command pixel_area)
  - (**pix_area_gamma0_geo**): illuminated area as obtained from integrating DEM-facets in gamma projection (command pixel_area)
  - (**pix_ratio_geo**): pixel area normalization factor (pix_ellip_sigma0 / pix_area_gamma0)
  - (**gs_ratio_geo**): gamma-sigma ratio (pix_gamma0 / pix_sigma0)
- additional files
  - **lut_init**: initial geocoding lookup table
- files specific to lookup table refinement
  - **lut_fine**: refined geocoding lookup table
  - **diffpar**: ISP offset/interferogram parameter file
  - **offs**: offset estimates (fcomplex)
  - **coffs**: culled range and azimuth offset estimates (fcomplex)
  - **coffsets**: culled offset estimates and cross correlation values (text format)
  - **ccp**: cross-correlation of each patch (0.0->1.0) (float)

Examples

geocode a Sentinel-1 scene and export the local incidence angle map with it

```
>>> from pyroSAR.gamma import geocode
>>> filename = 'S1A_IW_GRDH_1SDV_20180829T170656_20180829T170721_023464_028DE0_
↪F7BD.zip'
>>> geocode(scene=filename, dem='demfile', outdir='outdir', spacing=20, scaling=
↪'db',
>>>         export_extra=['dem_seg_geo', 'inc_geo', 'ls_map_geo'])
```

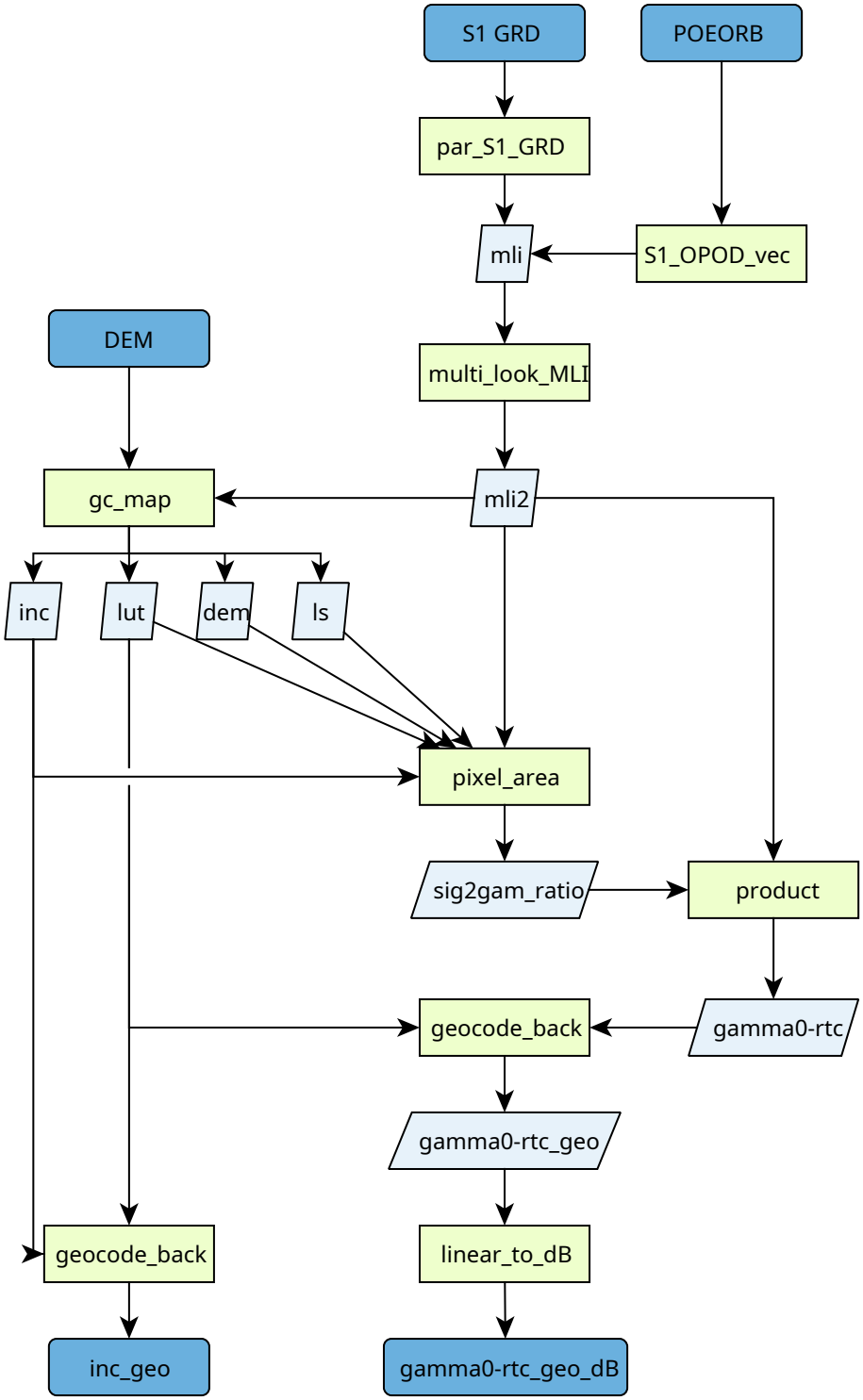

Fig. 3: Workflow diagram for function geocode for processing a Sentinel-1 Ground Range Detected (GRD) scene to radiometrically terrain corrected (RTC) gamma nought backscatter.

pyroSAR.gamma.**multilook**(*infile*, *outfile*, *spacing*, *rlks=None*, *azlks=None*, *exist_ok=False*, *logpath=None*, *outdir=None*, *shellscript=None*)

Multilooking of SLC and MLI images.

If the image is in slant range the ground range resolution is computed by dividing the range pixel spacing by the sine of the incidence angle.

The looks in range and azimuth are chosen to approximate the target resolution by rounding the ratio between target resolution and ground range/azimuth pixel spacing to the nearest integer.

An ENVI HDR parameter file is automatically written for better handling in other software.

Parameters

- **infile** (*str or list[str]*) – one of the following:
  - a SAR image in GAMMA format with a parameter file <infile>.par
  - a list of ScanSAR SLC swaths with parameter files <slc>.par and <slc>.tops_par; in this case a text file <outfile>_slc-tab.txt will be created, which is passed to the GAMMA command `multi_look_ScanSAR`
- **outfile** (*str*) – the name of the output GAMMA MLI file
- **spacing** (*int*) – the target pixel spacing in ground range
- **rlks** (*int or None*) – the number of range looks. If not None, overrides the computation done by function *pyroSAR.ancillary.multilook_factors()* based on the image pixel spacing and the target spacing.
- **azlks** (*int or None*) – the number of azimuth looks. Like *rlks*.
- **exist_ok** (*bool*) – allow existing output files and do not create new ones?
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the GAMMA commands to in shell format

See also:

pyroSAR.ancillary.multilook_factors

pyroSAR.gamma.**ovs**(*parfile*, *spacing*)

compute DEM oversampling factors for a target resolution in meters

Parameters

- **parfile** (*str*) – a GAMMA DEM parameter file
- **spacing** (*int or float*) – the target pixel spacing in meters

Returns

the oversampling factors for latitude and longitude

Return type

tuple of float

pyroSAR.gamma.**par2hdr**(*parfile*, *hdrfile*, *modifications=None*, *nodata=None*)

Create an ENVI HDR file from a GAMMA PAR file

Parameters

- **parfile** (*str*) – the GAMMA parfile
- **hdrfile** (*str*) – the ENVI HDR file
- **modifications** (*dict or None*) – a dictionary containing value deviations to write to the HDR file
- **nodata** (*int, float or None*) – a no data value to write to the HDR file via attribute 'data ignore value'

Examples

```
>>> from pyroSAR.gamma.auxil import par2hdr
>>> par2hdr('dem_seg.par', 'inc.hdr')
# write a HDR file for byte data based on a parfile of float data
>>> par2hdr('dem_seg.par', 'ls_map.hdr', modifications={'data_type': 1})
```

See also:

spatialist.envi.HDRobject, spatialist.envi.hdr()

pyroSAR.gamma.**process**(*cmd*, *outdir=None*, *logfile=None*, *logpath=None*, *inlist=None*, *void=True*, *shellscript=None*)

wrapper function to execute GAMMA commands via module subprocess

Parameters

- **cmd** (*list[str]*) – the command line arguments
- **outdir** (*str*) – the directory to execute the command in
- **logfile** (*str*) – a file to write the command log to; overrides parameter logpath
- **logpath** (*str*) – a directory to write logfiles to; the file will be named {GAMMA command}.log, e.g. gc_map.log; is overridden by parameter logfile
- **inlist** (*list*) – a list of values, which is passed as interactive inputs via stdin
- **void** (*bool*) – return the stdout and stderr messages?
- **shellscript** (*str*) – a file to write the GAMMA commands to in shell format

Returns
the stdout and stderr messages if void is False, otherwise None

Return type
tuple of str or None

2.3.2 DEM tools

A collection of functions to handle digital elevation models in GAMMA

dem_autocreate	automatically create a DEM in GAMMA format for a defined spatial geometry.
dem_import	convert an existing DEM in GDAL-readable format to GAMMA format including optional geoid-ellipsoid conversion.
dempar	create GAMMA parameter text files for DEM files
fill	interpolate missing values in the SRTM DEM (value -32768)
hgt	concatenate hgt file names overlapping with multiple SAR scenes
hgt_collect	automatic downloading and unpacking of srtm tiles
makeSRTM	Create a DEM in GAMMA format from SRTM tiles
mosaic	mosaicing of multiple DEMs
swap	byte swapping from small to big endian (as required by GAMMA)

pyroSAR.gamma.dem.**dem_autocreate**(*geometry*, *demType*, *outfile*, *buffer=None*, *t_srs=4326*, *tr=None*, *logpath=None*, *username=None*, *password=None*, *geoid_mode='gamma'*, *resampling_method='bilinear'*)

automatically create a DEM in GAMMA format for a defined spatial geometry.
The following steps will be performed:

- collect all tiles overlapping with the geometry using *pyroSAR.auxdata.dem_autoload()*
  - if they don't yet exist locally they will automatically be downloaded
  - the tiles will be downloaded into the SNAP auxdata directory structure, e.g. `$HOME/.snap/auxdata/dem/SRTM 3Sec`
- create a mosaic GeoTIFF of the same spatial extent as the input geometry plus a defined buffer using *pyroSAR.auxdata.dem_create()*
- if necessary, subtract the geoid-ellipsoid difference (see *pyroSAR.auxdata.dem_autoload()* for height references of different supported DEMs)
- convert the result to GAMMA format
  - If `t_srs` is *4326* and the DEM's height reference is either *WGS84* ellipsoid or *EGM96* geoid, the command `srtm2dem` can be used. This is kept for backwards compatibility.
  - For all other cases the newer command `dem_import` can be used if it exists and if the command `create_dem_par` accepts a parameter *EPSG*.

Parameters

- **geometry** (*spatialist.vector.Vector*) – a vector geometry delimiting the output DEM size
- **demType** (*str*) – the type of DEM to be used; see *dem_autoload()* for options
- **outfile** (*str*) – the name of the final DEM file
- **buffer** (*float or None*) – a buffer in degrees to create around the geometry
- **t_srs** (*int, str or osgeo.osr.SpatialReference*) – A target geographic reference system in WKT, EPSG, PROJ4 or OPENGIS format. See function `spatialist.auxil.crsConvert()` for details. Default: 4326.
- **tr** (*tuple or None*) – the target resolution as (xres, yres) in units of `t_srs`; if `t_srs` is kept at its default value of 4326, `tr` does not need to be defined and the original resolution is preserved; in all other cases the default of None is rejected
- **logpath** (*str*) – a directory to write GAMMA logfiles to
- **username** (*str or None*) – (optional) the user name for services requiring registration; see *dem_autoload()*
- **password** (*str or None*) – (optional) the password for the registration account
- **geoid_mode** (*str*) – the software to be used for converting geoid to ellipsoid heights (if necessary); options:
  - 'gamma'
  - 'gdal'
- **resampling_method** (*str*) – the gdalwarp resampling method; See here for options.

pyroSAR.gamma.dem.**dem_import**(*src*, *dst*, *geoid=None*, *logpath=None*, *outdir=None*)

convert an existing DEM in GDAL-readable format to GAMMA format including optional geoid-ellipsoid conversion.

Parameters

- **src** (*str*) – the input DEM
- **dst** (*str*) – the output DEM
- **geoid** (*str or None*) – the geoid height reference of *src*; supported options:
  - 'EGM96'
  - 'EGM2008'
  - None: assume WGS84 ellipsoid heights and do not convert heights
- **logpath** (*str or None*) – a directory to write logfiles to
- **outdir** (*str or None*) – the directory to execute the command in

pyroSAR.gamma.dem.**dempar**(*dem*, *logpath=None*)

create GAMMA parameter text files for DEM files

currently only EQA and UTM projections with WGS84 ellipsoid are supported

Parameters

- **dem** (*str*) – the name of the DEM
- **logpath** (*str*) – a directory to write logfiles to

pyroSAR.gamma.dem.**fill**(*dem*, *dem_out*, *logpath=None*, *replace=False*)

interpolate missing values in the SRTM DEM (value -32768)

Parameters

- **dem** (*str*) – the input DEM to be filled
- **dem_out** (*str*) – the name of the filled DEM
- **logpath** (*str*) – a directory to write logfiles to
- **replace** (*bool*) – delete *dem* once finished?

pyroSAR.gamma.dem.**hgt**(*parfiles*)

concatenate hgt file names overlapping with multiple SAR scenes

- this list is read for corner coordinates of which the next integer lower left latitude and longitude is computed
- hgt files are supplied in 1 degree equiangular format named e.g. N16W094.hgt (with pattern [NS][0-9]{2}[EW][0-9]{3}.hgt
- For north and east hemisphere the respective absolute latitude and longitude values are smaller than the lower left coordinate of the SAR image
- west and south coordinates are negative and hence the nearest lower left integer absolute value is going to be larger

Parameters
parfiles (*list of str or pyroSAR.ID*) – a list of GAMMA parameter files or pyroSAR ID objects

Returns
the names of hgt files overlapping with the supplied parameter files/objects

Return type
list

pyroSAR.gamma.dem.**hgt_collect**(*parfiles*, *outdir*, *demdir=None*, *arcsec=3*)

automatic downloading and unpacking of srtm tiles

Parameters

- **parfiles** (*list of str or pyroSAR.ID*) – a list of Gamma parameter files or pyroSAR ID objects
- **outdir** (*str*) – a target directory to download the tiles to
- **demdir** (*str or None*) – an additional directory already containing hgt tiles
- **arcsec** (*{1, 3}*) – the spatial resolution to be used

Returns

the names of all local hgt tiles overlapping with the parfiles

Return type

list

pyroSAR.gamma.dem.**makeSRTM**(*scenes*, *srtmdir*, *outname*)

Create a DEM in GAMMA format from SRTM tiles

- coordinates are read to determine the required DEM extent and select the necessary hgt tiles
- mosaics SRTM DEM tiles, converts them to GAMMA format and subtracts offset to WGS84 ellipsoid

intended for SRTM products downloaded from:

- USGS: https://gdex.cr.usgs.gov/gdex/
- CGIAR: https://srtm.csi.cgiar.org

Parameters

- **scenes** (*list of str or pyroSAR.ID*) – a list of Gamma parameter files or pyroSAR ID objects to read the DEM extent from
- **srtmdir** (*str*) – a directory containing the SRTM hgt tiles
- **outname** (*str*) – the name of the final DEM file

pyroSAR.gamma.dem.**mosaic**(*demlist*, *outname*, *byteorder=1*, *gammapar=True*)

mosaicing of multiple DEMs

Parameters

- **demlist** (*list[str]*) – a list of DEM names to be mosaiced
- **outname** (*str*) – the name of the final mosaic file
- **byteorder** (*{0, 1}*) – the byte order of the mosaic
  - 0: small endian
  - 1: big endian
- **gammapar** (*bool*) – create a GAMMA parameter file for the mosaic?

pyroSAR.gamma.dem.**swap**(*data*, *outname*)

byte swapping from small to big endian (as required by GAMMA)

Parameters

- **data** (*str*) – the DEM file to be swapped
- **outname** (*str*) – the name of the file to write

2.3.3 GAMMA Command API

This is an attempt to make it easier to execute GAMMA commands by offering automatically parsed Python functions. Thus, instead of executing the command via shell:

```
offset_fit offs ccp off.par coffs - 0.15 3 0 > offset_fit.log
```

one can wrap it in a Python script:

```
import os
from pyroSAR.gamma.api import isp

workdir = '/data/gamma_workdir'

parameters = {'offs': os.path.join(workdir, 'offs'),
              'ccp': os.path.join(workdir, 'ccp'),
              'OFF_par': os.path.join(workdir, 'off.par'),
              'coffs': os.path.join(workdir, 'coffs'),
              'thres': 0.15,
              'npoly': 3,
              'interact_flag': 0,
              'logpath': workdir}

isp.offset_fit(**parameters)
```

A file *offset_fit.log* containing the output of the command is written in both cases. Any parameters, which should not be written and need to be set to - in the shell can be omitted in the Python call since all optional parameters of the functions are already defined with '-' as a default. The documentation can be called like with any Python function:

```
from pyroSAR.gamma.api import isp
help(isp.offset_fit)
```

Parser Documentation

pyroSAR.gamma.parser.**autoparse**()

automatic parsing of GAMMA commands. This function will detect the GAMMA installation via environment variable *GAMMA_HOME*, detect all available modules (e.g. ISP, DIFF) and parse all the module's commands via function *parse_module()*. A new Python module will be created called *gammaparse*, which is stored under *$HOME/.pyrosar*. Upon importing the *pyroSAR.gamma* submodule, this function is run automatically and module *gammaparse* is imported as *api*.

Examples

```
>>> from pyroSAR.gamma.api import diff
>>> print('create_dem_par' in dir(diff))
True
```

pyroSAR.gamma.parser.**parse_command**(*command*, *indent=' '*)

Parse the help text of a GAMMA command to a Python function including a docstring. The docstring is in rst format and can thu be parsed by e.g. sphinx. This function is not intended to be used by itself, but rather within function *parse_module()*.

Parameters

- **command** (*str*) – the name of the gamma command
- **indent** (*str*) – the Python function indentation string; default: four spaces

Returns
the full Python function text

Return type
str

pyroSAR.gamma.parser.**parse_module**(*bindir*, *outfile*)

parse all Gamma commands of a module to functions and save them to a Python script.

Parameters

- **bindir** (*str*) – the *bin* directory of a module containing the commands
- **outfile** (*str*) – the name of the Python file to write

Examples

```
>>> import os
>>> from pyroSAR.gamma.parser import parse_module
>>> outname = os.path.join(os.environ['HOME'], 'isp.py')
>>> parse_module('/cluster/GAMMA_SOFTWARE-20161207/ISP/bin', outname)
```

API Demo

This is a demonstration of an output script as generated automatically by function *parse_module()* for the GAMMA module *ISP*. Within each function, the command name and all parameters are passed to function *process()*, which converts all input to str and then calls the command via the subprocess module.

pyroSAR.gamma.parser_demo.**ASAR_LO_phase_drift**(*SLC1_par*, *SLC2_par*, *OFF_par*, *ph_drift*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate interferometric phase correction due to drift of the ASAR local oscillator
Copyright 2015, Gamma Remote Sensing, v1.1 3-Dec-2015 clw

Parameters

- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **ph_drift** – (output) interferometric phase correction due to drift of the ASAR LO (radians)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ASAR_XCA**(*ASA_XCA*, *antenna*, *swath='-'*, *pol='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Interpretation of ASAR external calibration data file (ASA_XCA)
Copyright 2006, Gamma Remote Sensing, v1.1 7-June-2006 awi/uw/clw

Parameters

- **ASA_XCA** – (input) ASAR external calibration data file (binary)

- **antenna** –

  (output) 1-way antenna gain pattern file or '-' (if not provided)
  or 'all' to generate all ASAR antenna diagrams

- **swath** – ASAR swath (IS1,IS2,...IS7;SS1,SS2,...SS5)
- **pol** – polarization (HH,VV,HV,VH)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**DELFT_vec2**(*SLC_par*, *DELFT_dir*, *nstate='-'*, *interval='-'*, *ODR='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract and interpolate DELFT ERS-1, ERS-2, and ENVISAT state vectors
Copyright 2012, Gamma Remote Sensing, v2.6 clw 24-Oct-2012

Parameters

- **SLC_par** – (input) ISP image parameter file
- **DELFT_dir** –

  directory containing Delft orbit arclist and ODR files for ERS-1, ERS-2 or ENVISAT
  - NOTE: enter . for current directory

- **nstate** – number of state vectors to generate (enter - for default (>= 15)
- **interval** – time interval between state vectors in the ISP image parameter file (s) (default: 10.0)
- **ODR** – ODR file to use (include path) rather than ODR file determined from the Delft orbit arclist
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**DORIS_vec**(*SLC_par*, *DOR*, *nstate='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract ENVISAT DORIS state vectors and write to an ISP image parameter file
Copyright 2008, Gamma Remote Sensing, v1.4 11-Jun-2008 clw

Parameters

- **SLC_par** – (input/output)ISP SLC/MLI image parameter file
- **DOR** – (input) ASAR DORIS data file (DOR_VOR_AXVF)
- **nstate** – number of state vectors to extract (enter - for default: 11)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ERS_ASF_SLC**(*orbit*, *device*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **orbit** – orbit identifier (example: orbit number)
- **device** – EXABYTE tape device (no rewind); example: /dev/rmt/0mn
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ERS_ESA_PRI**(*orbit*, *device*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **orbit** – orbit identifier (example: orbit number)
- **device** – EXABYTE tape device (no rewind); example: /dev/rmt/0mn
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ERS_ESA_SLC**(*orbit*, *device*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **orbit** – orbit identifier (example: orbit number)
- **device** – EXABYTE tape device (no rewind); example: /dev/rmt/0mn
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**GRD_to_SR**(*GRD_par*, *MLI_par*, *OFF_par*, *in_file*, *out_file*, *rlks='-'*, *azlks='-'*, *interp_mode='-'*, *sr_rsp='-'*, *sr_azsp='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Conversion to slant range for ISP MLI and INSAR ground range data of type float
Copyright 2014, Gamma Remote Sensing, v2.0 4-Apr-2014 uw/clw

Parameters

- **GRD_par** – (input) SLC parameter file of output ground range image
- **MLI_par** –

  (output) MLI ISP image parameter file for slant range image (float)

  - NOTE: delete an existing MLI parameter file to recalculate the output MLI parameters
- **OFF_par** – (input) ISP offset/interferogram parameter file of input image (enter - image in MLI geometry)
- **in_file** – (input) ground range image (float)
- **out_file** – (output) slant range image (float)
- **rlks** – multi-looking in range (prior to resampling, default=1)
- **azlks** – multi-looking in azimuth (prior to resampling, default=1)
- **interp_mode** –

  interpolation mode

  - 0: nearest neighbor (default)
  - 1: spline
  - 2: spline log
- **sr_rsp** – output image slant range sample spacing (m) (default = c/(2*adc_sampling_rate)
- **sr_azsp** – output image azimunt sample spacing (m) (default = (input image azimuth spacing) * azlks)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**INTF_SLC**(*pass1*, *pass2*, *rlks*, *azlks*, *algorithm='-'*, *cc_win='-'*, *r_pos='-'*, *az_pos='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **pass1** – pass 1 identifier (example: pass number) reference
- **pass2** – pass 2 identifier (example: pass number)
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **algorithm** –

  algorithm used to determine offsets:
  1=intensity image cross correlation (default) 2=fringe visibility
- **cc_win** – window used for estimation of the correlation coefficient (default=3)
- **r_pos** – range position of center of image patch for initial offset
- **az_pos** – azimuth position of center of image patch for initial offset
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**MLI_cat**(*MLI_1*, *MLI_2*, *MLI1_par*, *MLI2_par*, *MLI_3*, *MLI3_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

Concatenate two MLI images using bicubic spline interpolation
Copyright 2015, Gamma Remote Sensing, v1.0 23-Jul-2015 awi

Parameters

- **MLI_1** – (input) MLI-1 image (single-look)
- **MLI_2** – (input) MLI-2 image to be appended to MLI-1
- **MLI1_par** – (input) MLI-1 ISP image parameter file
- **MLI2_par** – (input) MLI-2 ISP image parameter file
- **MLI_3** – (output) concatenated MLI image
- **MLI3_par** – (output) ISP image parameter file for concatenated image
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**MLI_copy**(*MLI_in*, *MLI_in_par*, *MLI_out*, *MLI_out_par*, *roff='-'*, *nr='-'*, *loff='-'*, *nl='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Copy MLI data file with options for segment extraction
Copyright 2013, Gamma Remote Sensing, v4.4 10-Jan-2013 uw/clw

Parameters

- **MLI_in** – (input) multi-look intensity image (float format)
- **MLI_in_par** – (input) ISP image parameter file for input MLI
- **MLI_out** – (output) selected MLI section (float format)
- **MLI_out_par** – (output) ISP image parameter file for output MLI
- **roff** – offset to starting range sample (enter - for default: 0)
- **nr** – number of range samples (enter - for default: to end of line
- **loff** – offset to starting line (enter - for default: 0)
- **nl** – number of lines to copy (enter - for default: to end of file)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**OPOD_vec**(*SLC_par*, *OPOD_dir*, *nstate='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/OPOD_vec
Copyright 2017, Gamma Remote Sensing, v1.2 7-Dec-2017 clw/awi
Extract Sentinel state vectors from an OPOD file and write these state vectors to an SLC parameter file
The required OPOD file located in a specified directory containing either restituted or precise state vectors

Parameters

- **SLC_par** – (input/output)ISP SLC/MLI image parameter file
- **OPOD_dir** –

  (input) directory containing Sentinel-1 precise or restituted OPOD orbit data file (AUX_POEORB or AUX_RESORB)
  https://qc.sentinel1.eo.esa.int/aux_resorb/
- **nstate** – number of state vectors to extract (default: include 60 sec extention at the start and end of the SLC data)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ORB_filt**(*SLC_par_in*, *SLC_par_out*, *interval='-'*, *extra='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Filter state vectors using a least-squares polynomial model
Copyright 2017, Gamma Remote Sensing, v1.1 21-Jun-2017 clw

Parameters

- **SLC_par_in** – (input) ISP image parameter file at least 5 state vectors
- **SLC_par_out** – (output) ISP image parameter file with state vectors filtered using least-squares
- **interval** – time interval between state vectors (enter - for default: state vector time interval in SLC_par)
- **extra** – extra time for state vectors at start and end of image (sec.) (enter - for default: 5.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ORB_prop_SLC**(*SLC_par*, *nstate='-'*, *interval='-'*, *extra='-'*, *mode='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate state vectors using orbit propagation and interpolation
Copyright 2008, Gamma Remote Sensing, v1.8 11-Jun-2008 clw/awi

Parameters

- **SLC_par** – (input) ISP image parameter file with at least 1 state vector
- **nstate** – number of state vectors to calculate (enter - for default: nstate from image duration + extra)
- **interval** – time interval between state vectors (enter - for default: state vector time interval in SLC_par)
- **extra** – extra time for state vectors at start and end of image (sec.) (enter - for default: 30.0)
- **mode** –

  orbit propagation mode:

  – 0: polynomial interpolation (default, if 3 or more state vectors available)

  - 1: integration of the equations of motion (default, if less than 3 state vectors available)
  - 2: interpolate between state vectors, minimum of 3 state vectors; interpolation of the equations of motion outside of the time span of the existing state vectors
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ORRM_vec**(*SLC_par*, *ORRM*, *nstate='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate state vectors extraction from ORRM file
Copyright 2008, Gamma Remote Sensing, v1.4 15-Nov-2004 clw

Parameters

- **SLC_par** – (input/output) ISP SLC/MLI image parameter file
- **ORRM** – (input) ORRM state vector file
- **nstate** – number of state vectors (default=5, maximum=1024)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**PRC_vec**(*SLC_par*, *PRC*, *nstate='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

State vectors from ERS PRC orbit data for ISP processing clw/uw
Copyright 2008, Gamma Remote Sensing, v1.7 clw 11-Jun-2008

Parameters

- **SLC_par** – (input/output) ISP SLC/MLI image parameter file
- **PRC** – (input) PRC state vector file
- **nstate** – number of state vectors (default=5, maximum=1024)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**RSAT2_vec**(*SLC_par*, *RSAT2_orb*, *nstate='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract Radarsat-2 state vectors from a definitive orbit file
Copyright 2010, Gamma Remote Sensing, v1.0 clw 13-May-2010

Parameters

- **SLC_par** – (input) ISP image parameter file
- **RSAT2_orb** – Radarsat-2 definitive orbit data file available from MDA. (orbit_number_def.orb)

- **nstate** – number of state vectors to extract (enter - for default: 9)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_BURST_tab**(*SLC1_tab*, *SLC2_tab*, *BURST_tab*, *logpath=None*, *outdir=None*, *shellscript=None*)

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/S1_BURST_tab

Copyright 2017, Gamma Remote Sensing, v1.0 25-May-2017 clw

Calculate Sentinel BURST_tab based on parameters extracted from SLC parameter files listed in SLC1_tab and SLC2_tab

Running SLC_copy_S1_TOPS using BURST_tab will generate SLC-2 data with matching bursts for each swath of SLC-1 and SLC-2

Parameters

- **SLC1_tab** – (input) 3 column list of the reference TOPS SLC swaths in row order IW1, IW2, IW3
- **SLC2_tab** – (input) 3 column list of TOPS SLC-2 swaths to be resampled to the geometry of the reference SLC1 in row order IW1, IW2, IW3.
- **BURST_tab** – (output) 2 column list of the first and last bursts to copy from each swath, one line for each swath
- **BURST_tab** – line entries: first_burst last_burst Note: first burst is 1
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_BURST_tab_from_zipfile**(*zipfile_list*, *zipfile_ref*, *burst_number_table_ref='-'*, *cleaning='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **zipfile_list** –

  (input) ASCII file containing S1 zip filename(s) of one data take
  indicate - to generate burst_number_table of reference TOPS SLC
- **zipfile_ref** – (input) S1 zip filename for the reference TOPS SLC
- **burst_number_table_ref** –

  (input) ASCII file containing first/last burst numbers selected
  indicate - to use all bursts as present in the reference TOPS SLC zipfile
- **cleaning** –

  flag to indicate if intermediate files are deleted (default=1: yes, 0: not deleted)
  intermediate and output filenames are generated based on the zip file names
- **logpath** (*str or None*) – a directory to write command logfiles to

- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_GRD_preproc**(*S1_list*, *MLI_dir*, *pol*, *log*, *logpath=None*, *outdir=None*, *shellscript=None*)

Preprocessing of Sentinel-1 TOPS GRD products, extract GRD data and generate MLI prodcuts
Copyright 2018, Gamma Remote Sensing, v1.2 19-Mar-2018 clw/cm

Parameters

- **S1_list** – (input) single column text file. Entries are directories (including path) containing Sentinel-1 TOPS GRD products
- **MLI_dir** –

  directory for output SLC data files and SLC parameter files

  - NOTE: output file names have the form : 20150119_hh.mli
- **pol** – SLC polarization to extract (hh,hv,vh,vv)
- **log** –

  (output) S1 GRD pre-processing log file

-c	(option) apply radiometric calibration factor without noise subtraction
-n	(option) apply radiometric calibration factor with noise subtraction
-t	(option) include full timestamp YYYYMMD-DtHHMMSSin SLC and SLC_par filenames, default YYYYMMDD

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_OPOD_vec**(*SLC_par*, *OPOD*, *nstate='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract Sentinel-1 OPOD state vectors and copy into the ISP image parameter file
Copyright 2017, Gamma Remote Sensing, v1.3 09-Mar-2017 awi/clw

Parameters

- **SLC_par** – (input/output)ISP SLC/MLI image parameter file
- **OPOD** –

  (input) Sentinel-1 OPOD orbit data file (AUX_POEORB or AUX_RESORB)
  https://qc.sentinel1.eo.esa.int/aux_resorb/
- **nstate** – number of state vectors to extract (default: include 60 sec extention at the start and end of the SLC data)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_TOPS_preproc**(*S1_list*, *SLC_dir*, *pol*, *log*, *logpath=None*, *outdir=None*, *shellscript=None*)

Preprocessing of Sentinel-1 TOPS SLC products, extract SLC data and generate SLC_tab
Copyright 2018, Gamma Remote Sensing, v2.2 10-Jan-2018 clw/awi

Parameters

- **S1_list** – (input) single column text file. Enteries are directories (including path) containing Sentinel-1 TOPS SLC products
- **SLC_dir** –

  directory for output SLC data files and SLC parameter files
  Note: output file names have the form : 20150119_iw1_hh.slc
- **pol** – SLC polarization to extract (hh,hv,vh,vv)
- **log** –

  (output) S1 SLC pre-processing log file

-c	(option) apply radiometric calibration factor without noise subtraction
-n	(option) apply radiometric calibration factor with noise subtraction
-s	(option) output is SCOMPLEX format (default: FCOMPLEX)
-t	(option) include full timestamp YYYYMMDD-DtHHMMSS in SLC and SLC_par filenames, default YYYYMMDD
-m MLI_dir	(option) calculate MLI images and store in MLI_dir, enter . for current directory
-r rlks	(option) number of MLI range looks (default: 10)
-a azlks	(option) number of MLI azimuth looks (default: 2)
-b SLC_tab	(option) SLC_tab filename, by default SLC_tab_YYMMDD or SLC_tab_YYYYMMDDtHHMMSS
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_burstloc**(*annotation_XML*, *logpath=None*, *outdir=None*, *shellscript=None*)

Print Burst information found in the Sentinel-1 annotation file
Copyright 2018, Gamma Remote Sensing, v1.1 7-Feb-2018 awi/cm

Parameters

- **annotation_XML** – (input) Sentinel-1 L1 XML annotation file
- **logpath** (*str or None*) – a directory to write command logfiles to

- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_extract_png**(*zipfile*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **zipfile** – (input) Sentinel-1 zipfile (GRD or SLC)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**S1_import_SLC_from_zipfiles**(*zipfile_list*, *burst_number_table_ref='-'*, *pol='-'*, *dtype='-'*, *swath_flag='-'*, *cleaning='-'*, *noise_mode='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **zipfile_list** – (input) ASCII file containing S1 zip filename(s) of one data take (one per line, in correct sequence)
- **burst_number_table_ref** –

  (input) ASCII file containing first/last burst numbers selected
  indicate - to use all bursts present in the indicated zipfiles
- **pol** –

  polarization flag (default: -, other values are vh, hh, hv)
  indicate - to use all polarizations available in the indicated zipfiles
- **dtype** – output data type: default=0 (FCOMPLEX), 1: SCOMPLEX
- **swath_flag** – flag to select sub-swaths to read (default=0 (as listed in burst_number_table_ref, all if no
- **burst_number_table_ref** –

  provided), 1,2,3 (1 sub-swath only), 4 (1&2), 5 (2&3)
  OPOD_dir directory with OPOD state vector files (default: .)
- **cleaning** – flag to indicate if intermediate files are deleted (default = 1 –> deleted, 0: not deleted)
- **noise_mode** –

  noise mode (default=1: apply noise correction; 2: do not apply and write out noise file
  resulting files: burst SLC files (per polarization, with SLC_tab, SLC, SLC_par, TOPS_par and optionally SLC.noise) (concatenated, empty bursts added where necessary) at selected polarizations
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SBI_INT**(*RSLC1*, *RSLC_par1*, *RSLC2*, *RSLC_par2*, *sbi*, *off*, *sbi_pwr*, *par_out*, *norm_sq='-'*, *rlks='-'*, *azlks='-'*, *iwflg='-'*, *cflg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **RSLC1** – (input) master single-look complex image (fcomplex or scomplex)
- **RSLC_par1** – (input) SLC ISP image parameter file of RSLC1
- **RSLC2** – (input) co-registered slave SLC image (fcomplex or scomplex)
- **RSLC_par2** – (input) SLC ISP image parameter file of RSLC2
- **sbi** – (output) multi-look split-beam interferogram (fcomplex)
- **off** – (output) ISP offset parameter file for multi-look split-beam interferogram (ascii)
- **sbi_pwr** – (output) multi-look reference backscatter intensity image (float)
- **par_out** – (output) SLC/MLI ISP image parameter file of sbi_pwr
- **norm_sq** – normalized squint difference parameter (default: 0.5)
- **rlks** – number of range looks in output split-beam interferogram (default: 1)
- **azlks** – number of azimuth looks in output split-beam interferogram (default: 1)
- **iwflg** –

  inverse weighting flag:

  - 0: do not remove azimuth processing spectral window (default)
  - 1: apply inverse of azimuth compression processing window
- **cflg** –

  flag to indicate if intermediate data (e.g. filtered slc) are deleted

  - 0: intermediate data are deleted (default)
  - 1: intermediate data are NOT deleted file names for band-pass filtered SLC are generated automatically by adding the letter b / f for the backward / foward looking beam
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_adf**(*SLC*, *ref_SLC*, *ref_SLC_par*, *SLC_filt*, *mode='-'*, *alpha='-'*, *nfft_r='-'*, *nfft_az='-'*, *r_step='-'*, *az_step='-'*, *mwin_r='-'*, *mwin_az='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Adaptive filtering of SLC data based on the local PSD of a reference SLC image
Copyright 2017, Gamma Remote Sensing, v1.2 29-Sep-2017 clw

Parameters

- **SLC** – (input) SLC to be filtered (FCOMPLEX or SCOMPLEX)
- **ref_SLC** – (input) reference SLC
- **ref_SLC_par** – (input) reference SLC parameter file

- **SLC_filt** – (output) output filtered SLC using the power spectrum of the reference SLC
- **mode** –

  SLC filtering mode (enter - for default):

  - 0: 1D range PSD filter
  - 1: 1D azimuth PSD filter
  - 2: 2D range PSD * azimuth PSD filter
  - 3: 2D median-filtered PSD filtering (default)
- **alpha** – exponent to apply to PSD value (enter - for default: 0.30)
- **nfft_r** – range filter FFT window size, 2**N, 16->1024, (enter - for default: 128)
- **nfft_az** – azimuth filter FFT window size, 2**N, 16->1024, (enter - for default: 128)
- **r_step** – range processing step (enter - for default: nfft_r/4)
- **az_step** – azimuth processing step (enter - for default: nfft_az/4)
- **mwin_r** – range median window size for median PSD filtering (enter - for default: 5)
- **mwin_az** – azimuth median window size for median PSD filtering (enter - for default: 5)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_burst_copy**(*SLC*, *SLC_par*, *TOPS_par*, *SLC_out*, *SLC_out_par*, *burst_num*, *drflg='-'*, *SLC_par2='-'*, *dtype='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Copy selected burst from Sentinel-1 TOPS SLC to a file
Copyright 2017, Gamma Remote Sensing, v1.4 24-Nov-2017 awi/clw/cm

Parameters

- **SLC** – (input) Sentinel-1 TOPS mode burst SLC
- **SLC_par** – (input) SLC parameter file for the TOPS burst SLC
- **TOPS_par** – (input) TOPS parameter file for the TOPS burst SLC
- **SLC_out** – (output) SLC file containing a single burst
- **SLC_out_par** – (output) SLC parameter file for the single burst SLC
- **burst_num** – burst number of selected burst (1->number of bursts in the SLC)
- **drflg** –

  deramp phase flag (enter - for default)

  - 0: no modification of the burst SLC phase (default)
  - 1: subtract TOPS Doppler phase ramp (deramp)
- **SLC_par2** – (output) SLC parameter file for the single burst SLC with deramped phase (drflg: 1, enter - for none)
- **dtype** –

  output data type (enter - for default: same as input data):

  - 0: FCOMPLEX

  - 1: SCOMPLEX
  - NOTE: the program also supports FLOAT data; no data type conversion from or to FLOAT is possible
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_burst_corners**(*SLC_par*, *TOPS_par*, *kml='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate corner geographic coordinates of Sentinel-1 TOPS SLC bursts
Copyright 2017, Gamma Remote Sensing, v1.1 22-Aug-2017 awi/rc/cw

Parameters

- **SLC_par** – (input) SLC parameter file for the TOPS burst SLC
- **TOPS_par** – (input) TOPS parameter file for the TOPS burst SLC
- **kml** – (output) kml output file
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_cat**(*SLC_1*, *SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *SLC_3*, *SLC3_par*, *dopflg='-'*, *iflg='-'*, *phflg='-'*, *gainflg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Concatenate two SLC images using 2-D SINC interpolation
Copyright 2018, Gamma Remote Sensing, v1.9 30-Apr-2018 clw/cm

Parameters

- **SLC_1** – (input) SLC-1 image
- **SLC_2** – (input) SLC-2 image to be appended to SLC-1
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset parameter file containing offset polynomials between SLC-1 and SLC-2
- **SLC_3** – (output) concatenated SLC
- **SLC3_par** – (output) ISP image parameter file for concatenated image
- **dopflg** –

  Doppler flag: (enter - for default)

  - 0: ignore Doppler centroid information, assume 0 Hz Doppler centroid
  - 1: use Doppler centroid information for interpolation (default)
- **iflg** –

  input data type flag: (enter - for default)

  - 0: input data are SLC images, use data type specified in SLC_par files (SCOMPLEX or FCOMPLEX) (default)
  - 1: input scenes are interferograms, force FCOMPLEX data type

- **phflg** –

  phase offset correction flag: (enter - for default)

  - 0: no phase offset correction for SLC-2 (default)
  - 1: apply phase offset correction to SLC-2

- **gainflg** –

  gain correction flag: (enter - for default)

  - 0: no gain correction for SLC-2 (default)
  - 1: apply gain correction to SLC-2

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_cat_S1_TOPS**(*SLC_tab1*, *SLC_tab2*, *SLC_tab3*, *logpath=None*, *outdir=None*, *shellscript=None*)

Concatenate adjacent Sentinel-1 TOPS SLC images
Copyright 2018, Gamma Remote Sensing v2.2 8-May-2018 clw/cm

Parameters

- **SLC_tab1** –

  (input) 3 column list of the reference TOPS SLC swaths in row order IW1, IW2, IW3... (earlier time)
  SLC_tab line entries: SLC SLC_par TOPS_par

- **SLC_tab2** – (input) 3 column list of TOPS SLC-2 swaths in the same order as the SLC_tab1 IW1, IW2, IW3... (later time)
- **SLC_tab3** – (input) 3 column list of the output concatenated TOPS swaths in the order IW1, IW2, IW3...
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_copy**(*SLC_in*, *SLC_par_in*, *SLC_out*, *SLC_par_out*, *fcase='-'*, *sc='-'*, *roff='-'*, *nr='-'*, *loff='-'*, *nl='-'*, *swap='-'*, *header_lines='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Copy SLC with options for data format conversion, segment extraction, and byte swapping
Copyright 2015, Gamma Remote Sensing, v5.1 13-Aug-2015 uw/clw

Parameters

- **SLC_in** – (input) SLC (FCOMPLEX or SCOMPLEX format)
- **SLC_par_in** – (input) ISP SLC parameter file for input SLC
- **SLC_out** – (output) selected SLC section (FCOMPLEX or SCOMPLEX format)

- **SLC_par_out** – (output) ISP SLC parameter file of output SLC
- **fcase** –

  data format conversion (enter - for default: output format = input format)

  - 1: FCOMPLEX –> FCOMPLEX (default sc = 1.0)
  - 2: FCOMPLEX –> SCOMPLEX (default sc = 10000.0)
  - 3: SCOMPLEX –> FCOMPLEX (default sc = 0.0001)
  - 4: SCOMPLEX –> SCOMPLEX (default sc = 1.0)
- **sc** – scale factor for input SLC data (enter - for default)
- **roff** – offset to starting range sample (enter - for default: 0)
- **nr** – number of range samples (enter - for default: to end of line)
- **loff** – offset to starting line (enter - for default: 0)
- **nl** – number of lines to copy (enter - for default: to end of file)
- **swap** –

  swap data (enter - for default)

  - 0: normal (default)
  - 1: swap real/imaginary part of complex data
  - 2: swap left/right (near/far range)
- **header_lines** –

  number of input file header lines (enter - for default: 0)

  - NOTE: CEOS format SLC data have 1 header line
  - NOTE: file offset pointer size (bytes): 8
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_copy_S1_TOPS**(*SLC1_tab*, *SLC2_tab*, *BURST_tab*, *dtype='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Copy multiple bursts from a Sentinel-1 TOPS SLC to an output TOPS SLC
Copyright 2018, Gamma Remote Sensing v2.4 25-Apr-2018 clw/cm

Parameters

- **SLC1_tab** –

  (input) 3 column list of TOPS SLC-1 swaths to be copied in row order IW1, IW2, IW3:
  SLC_tab line entries: SLC SLC_par TOPS_par
- **SLC2_tab** – (input) 3 column list of the output copied SLC-1 TOPS swaths in the order IW1, IW2, IW3
- **BURST_tab** – (input) 2 column list of the first and last burst to copy from each swath, one line for each swath
- **BURST_tab** –

line entries: first_burst last_burst Note: first burst is 1, enter - to select last physical burst

Note: if first_burst <= 0, then blank bursts are generated at the start of the output swath if last_burst exceeds the number of bursts in the input data swath, then blank bursts are appended to the end of the output swath

- **dtype** –

  output data type (enter - for default: same as input data):

  - 0: FCOMPLEX
  - 1: SCOMPLEX
  - NOTE: the program also supports FLOAT data; no data type conversion from or to FLOAT is possible

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_copy_WB**(*SLC_tab*, *SLC2_dir*, *logpath=None*, *outdir=None*, *shellscript=None*)

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/SLC_copy_WB
Copyright 2011, Gamma Remote Sensing, v1.1 9-Apr-2011 clw
Create a new set of SLCs for all beams in a PALSAR WB ScanSAR image

Parameters

- **SLC_tab** – (input) two column list of input SLC files and SLC ISP image parameter files (including paths) (text)
- **SLC2_dir** –

  directory to contain copied segments of the input SLC data and the associated parameter files

  - NOTE: current directory is denoted using .

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_corners**(*SLC_par*, *terra_alt='-'*, *kml='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate SLC/MLI image corners in geodetic latitude and longitude (deg.)
Copyright 2017, Gamma Remote Sensing, v1.8 13-Dec-2017 clw/awi/cm

Parameters

- **SLC_par** – (input) ISP SLC/MLI image parameter file
- **terra_alt** – (input) average terrain altitude (enter - for default: 300.000 meters)
- **kml** – (output) kml output file (enter - for none)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_deramp**(*SLC_1*, *SLC_par1*, *SLC_2*, *SLC_par2*, *mode*, *dop_ph='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate and subtract Doppler phase from an SLC image
Copyright 2016, Gamma Remote Sensing, v1.5 4-Feb-2016 clw

Parameters

- **SLC_1** – (input) SLC data file (fcomplex or scomplex format)
- **SLC_par1** – (input) SLC parameter file with Doppler information
- **SLC_2** – (output) SLC with Doppler phase removed (or added)
- **SLC_par2** – (output) SLC parameter file for the output SLC
- **mode** –

  mode of operation:

  - 0: subtract Doppler phase ramp (deramp)
  - 1: add Doppler phase ramp (reramp)
- **dop_ph** –

  (output) Doppler phase (FLOAT)
  Note: SLC_par1 contains the Doppler polynomial that is used to calculate the Doppler phase ramp
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_deramp_S1_TOPS**(*SLC1_tab*, *SLC2_tab*, *mode*, *phflg*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate and subtract S1 TOPS Doppler phase from burst SLC data
Copyright 2018, Gamma Remote Sensing v1.6 25-Apr-2018 clw/cm

Parameters

- **SLC1_tab** –

  (input) 3 column list of TOPS SLC-1 swaths to be deramped in row order IW1, IW2, IW3:
  SLC_tab line entries: SLC SLC_par TOPS_par
- **SLC2_tab** – (input) 3 column list of the output deramped SLC-1 TOPS swaths in the order IW1, IW2, IW3
- **mode** –

  mode of operation:

  - 0: subtract TOPS Doppler phase (deramp)
  - 1: add Doppler phase ramp (reramp)
- **phflg** –

  deramp phase flag:

  - 0: do not save TOPS Doppler phase (default)

  - 1: save TOPS Doppler phase, output filename is the same as the deramped SLC with extension .dph
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_freq_shift**(*SLC*, *SLC_par*, *SLC_shift*, *SLC_shift_par*, *freq_shift*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/SLC_freq_shift.c
Copyright 2017, Gamma Remote Sensing, v1.0 28-Sep-2017 clw
Shift the effective radar carrier frequency of an SLC image by a specified amount

Parameters

- **SLC** – (input) SLC file (FCOMPLEX or SCOMPLEX)
- **SLC_par** – (input) SLC parameter file
- **SLC_shift** – (output) SLC data with shifted radar carrier frequency
- **SLC_shift_par** – (output) SLC parameter file with shifted radar carrier frequency
- **freq_shift** – radar carrier frequency shift (Hz)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_interp**(*SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *SLC_2R*, *SLC2R_par*, *loff='-'*, *nlines='-'*, *mode='-'*, *order='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

SLC complex image resampling using 2-D Lanczos or B-spline interpolation
Copyright 2017, Gamma Remote Sensing, v4.6 4-Dec-2017 clw/cm

Parameters

- **SLC_2** – (input) SLC-2 image to be resampled to the geometry of the SLC-1 reference image
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **SLC_2R** – (output) single-look complex image 2 coregistered to SLC-1
- **SLC2R_par** – (output) SLC-2R ISP image parameter file for coregistered image
- **loff** – offset to first valid output line (in SLC-1 lines) (enter - for default: 0)
- **nlines** – number of valid output lines (enter - or 0 for default: to end of file)
- **mode** –

  interpolation mode (enter - for default)

  - 0: Lanczos (default)
  - 1: B-spline

- **order** – Lanczos interpolator order / B-spline degree 4 -> 9 (enter - for default: 4)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_interp_S1_TOPS**(*SLC2_tab*, *SLC2_par*, *SLC1_tab*, *SLC1_par*, *OFF_par*, *SLC2R_tab*, *SLC_2R='-'*, *SLC2R_par='-'*, *mode='-'*, *order='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Resample S1 TOPS (IW mode) SLC using global offset polynomial
Copyright 2018, Gamma Remote Sensing v2.4 25-Apr-2018 clw/cm

Parameters

- **SLC2_tab** –

  (input) 3 column list of TOPS SLC-2 swaths to be resampled to the geometry of the reference SLC1 in row order IW1, IW2, IW3:
  SLC_tab line entries: SLC SLC_par TOPS_par
- **SLC2_par** – (input) SLC parameter file of TOPS SLC-2 mosaic, SLC-2 is generated from the TOPS swaths listed in SLC2_tab
- **SLC1_tab** – (input) 3 column list of the reference TOPS SLC swaths in row order IW1, IW2, IW3
- **SLC1_par** – (input) SLC parameter file of the reference TOPS SLC-1 mosaic, SLC-1 is generated from the TOPS swaths listed in SLC1_tab
- **OFF_par** – (input) global ISP offset and interferogram parameter file, the offset model is determined from the TOPS SLC mosaics
- **SLC2R_tab** – (input) 3 column list of the output resampled SLC-2 TOPS swaths in the order IW1, IW2, IW3
- **SLC_2R** – (output) resampled mosaic generated from the swaths listed in SLC2R_tab, coregisted to the TOPS SLC-1 mosaic (enter - for none)
- **SLC2R_par** – (output) SLC parameter file associated with the resampled TOPS SLC-2R mosaic (enter - for none)
- **mode** –

  interpolation mode (enter - for default)

  - 0: Lanczos (default)
  - 1: B-spline
- **order** – Lanczos interpolator order / B-spline degree 4 -> 9 (enter - for default: 4)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_interp_map**(*SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *SLC_2R*, *SLC2R_par*, *OFF_par2*, *coffs_sm*, *loff='-'*, *nlines='-'*, *mode='-'*, *order='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

SLC image resampling using a 2-D offset map
Copyright 2017, Gamma Remote Sensing, v3.9 4-Dec-2017 clw/uw/cm

Parameters

- **SLC_2** – (input) SLC-2 image to be resampled to the reference SLC-1 reference image
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **SLC_2R** – (output) single-look complex image 2 coregistered to SLC-1
- **SLC2R_par** – (output) SLC-2R ISP image parameter file for co-registered image
- **OFF_par2** – (input) ISP offset/interferogram parameter file used for residual offsets map (coffs_sm)
- **coffs_sm** – (input) smoothed residual range and azimuth offsets (fcomplex)
- **loff** – offset to first valid output line (in SLC-1 lines) (enter - for default: 0)
- **nlines** – number of valid output lines (enter - or 0 for default: to end of file)
- **mode** –

  interpolation mode (enter - for default)

  - 0: Lanczos (default)
  - 1: B-spline
- **order** – Lanczos interpolator order / B-spline degree 4 -> 9 (enter - for default: 4)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_intf**(*SLC_1*, *SLC_2R*, *SLC1_par*, *SLC2R_par*, *OFF_par*, *interf*, *rlks*, *azlks*, *loff='-'*, *nlines='-'*, *sps_flg='-'*, *azf_flg='-'*, *rp1_flg='-'*, *rp2_flg='-'*, *SLC_1s='-'*, *SLC_2Rs='-'*, *SLC_1s_par='-'*, *SLC_2Rs_par='-'*, *az_beta='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Interferogram generation from co-registered SLC data
Copyright 2017, Gamma Remote Sensing, v5.5 clw/uw/cm 26-Aug-2017

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2R** – (input) single-look complex image 2 coregistered to SLC-1
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2R_par** – (input) SLC-2R ISP image parameter file for the co-registered image
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **interf** – (output) interferogram from SLC-1 and SLC-2R
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **loff** – offset to starting line relative to SLC-1 for interferogram (default=0)

- **nlines** – number of SLC lines to process (enter - for default: to end of file)
- **sps_flg** –

  range spectral shift flag:

  - 1: apply range spectral shift filter (default)
  - 0: do not apply range spectral shift filter
- **azf_flg** –

  azimuth common band filter flag:

  - 1: apply azimuth common-band filter (default)
  - 0: do not apply azimuth common band filter
- **rp1_flg** –

  SLC-1 range phase mode

  - 0: nearest approach (zero-Doppler) phase
  - 1: ref. function center (Doppler centroid) (default)
- **rp2_flg** –

  SLC-2 range phase mode

  - 0: nearest approach (zero-Doppler) phase
  - 1: ref. function center (Doppler centroid) (default)
- **SLC_1s** – SLC-1 after range spectral shift and azimuth common-band filtering (FCOMPLEX format) (enter - for none)
- **SLC_2Rs** – SLC-2R after range spectral shift and azimuth common-band filtering (FCOMPLEX format) (enter - for none)
- **SLC_1s_par** – SLC-1s ISP image parameter file (enter - for none)
- **SLC_2Rs_par** – SLC-2Rs ISP image parameter file (enter - for none)
- **az_beta** – azimuth common-band filter Kaiser window parameter (default: 2.120)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_mosaic_S1_TOPS**(*SLC_tab*, *SLC*, *SLC_par*, *rlks*, *azlks*, *wflg='-'*, *SLCR_tab='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate SLC mosaic of Sentinel-1 TOPS burst SLC data
Copyright 2018, Gamma Remote Sensing v3.6 25-Apr-2018 clw/awi/cm

Parameters

- **SLC_tab** – (input) 3 column list of SLC, SLC_par, Sentinel-1 TOPS_par sorted in the order IW1, IW2, IW3…
- **SLC** – (output) SLC mosaic image
- **SLC_par** – (output) SLC mosaic image parameter file
- **rlks** – number of range looks used to determine burst window boundaries for the mosaic
- **azlks** – number of azimuth looks used to determine burst window boundaries for the mosaic

- **wflg** –

  burst window calculation flag:

  - 0: use existing burst window parameters if they exist, otherwise calculate burst window parameters (default)
  - 1: calculate burst window parameters from burst parameters and the number of range and azimuth looks

- **SLCR_tab** –

  (input) SLC_tab of the reference scene, 3 column list of SLC, SLC_par, TOPS_par sorted sorted in the order IW1, IW2, IW3

  - NOTE: When generating a mosaic of a resampled SLC, the SLC_tab of the reference scene is required

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_ovr**(*SLC*, *SLC_par*, *SLC_ovr*, *SLC_ovr_par*, *r_ovr*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/SLC_ovr.c
Copyright 2017, Gamma Remote Sensing, v1.9 12-Oct-2017 clw/cm
Oversample or subsample SLC data in slant-range

Parameters

- **SLC** – (input) SLC file (fcomplex or scomplex)
- **SLC_par** – (input) SLC parameter file of SLC file
- **SLC_ovr** – (output) range resampled SLC file (fcomplex or scomplex)
- **SLC_ovr_par** – (output) SLC parameter file of range resampled SLC data file
- **r_ovr** –

  integer range oversampling factor (2 –> 16)
  if r_ovr < 0, the SLC will be subsampled, integer range subsampling factor (-2 –> -16)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_ovr2**(*SLC*, *SLC_par*, *SLC_ovr*, *SLC_ovr_par*, *r_ovr*, *az_ovr='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **SLC** – (input) SLC file (SCOMPLEX or FCOMPLEX, e.g. 20141126.SLC)
- **SLC_par** – (input) SLC parameter file (e.g. 20141126.SLC.par)
- **SLC_ovr** – (output) oversampled SLC file (same format as SLC)
- **SLC_ovr_par** – (output) SLC parameter file of oversampled SLC file
- **r_ovr** – range oversampling factor (e.g. 2.5)

- **az_ovr** – azimuth oversampling factor (e.g. 2.5, default = 1.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SLC_phase_shift**(*SLC_1*, *SLC_par1*, *SLC_2*, *SLC_par2*, *ph_shift*, *logpath=None*, *outdir=None*, *shellscript=None*)

Add a constant phase from an SLC image
Copyright 2015, Gamma Remote Sensing, v1.1 1-Dec-2015 clw

Parameters

- **SLC_1** – (input) SLC data file (fcomplex or scomplex format)
- **SLC_par1** – (input) SLC parameter file
- **SLC_2** – (output) SLC with phase shift
- **SLC_par2** – (output) SLC parameter file for the output SLC
- **ph_shift** –

  phase shift to add to SLC phase (radians)

  - NOTE: Used to apply a constant phase shift of -1.25 radians to Sentinel-1 TOPS SLC data from swath IW1 acquired up to 10-Mar-2015. Used to apply a constant phase shift of -3.83 radians to Sentinel-1 TOPS SLC data with H-POL on receive (e.g. VH) acquired up to 10-Mar-2015.
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**SR_to_GRD**(*MLI_par*, *OFF_par*, *GRD_par*, *in_file*, *out_file*, *rlks='-'*, *azlks='-'*, *interp_mode='-'*, *grd_rsp='-'*, *grd_azsp='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Conversion to ground range for ISP MLI and INSAR data of type float
Copyright 2009, Gamma Remote Sensing, v1.9 7-May-2009 uw/clw

Parameters

- **MLI_par** – (input) MLI image parameter file of input slant range image (float)
- **OFF_par** – (input) ISP offset/interferogram parameter file of input image (enter - image in MLI geometry)
- **GRD_par** – (input/output) image parameter file of output ground range image
- **in_file** – (input) slant range image (float)
- **out_file** – (output) ground range image (float)
- **rlks** – multi-looking in range (prior to resampling, default=1)
- **azlks** – multi-looking in azimuth (prior to resampling, default=1)
- **interp_mode** –

  interpolation mode

  - 0: nearest neighbor (default)

  - 1: spline
  - 2: spline log
- **grd_rsp** – output image ground range sample spacing (m) (default = (input image azimuth spacing) * azlks)
- **grd_azsp** – output image azimuth sample spacing (m) (default = (input image azimuth spacing) * azlks)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**TX_SLC_preproc**(*TSX_list*, *SLC_dir*, *log*, *logpath=None*, *outdir=None*, *shellscript=None*)

Preprocessing of TerraSAR-X TDX1 and TSX1 SLC products using par_TX_SLC
Copyright 2013, Gamma Remote Sensing, v1.2 22-Oct-2013 clw

Parameters

- **TSX_list** –

  (input) single column text file with directories (including path)
  containing path to directory containing product XML for IMAGEDATA/*.cos files
- **SLC_dir** – directory for output SLC data files and SLC parameter files
- **log** – (output) processing log file
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**UNWRAP**(*interf*, *cc*, *pwr*, *unwrap*, *flag*, *width*, *lines*, *corr_thr='-'*, *pwr_thr='-'*, *r_init='-'*, *az_init='-'*, *r1='-'*, *r2='-'*, *l1='-'*, *l2='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **interf** – interferogram filename (*.int, *.flt)
- **cc** – correlation filename (*.cc)
- **pwr** – intensity image (*.pwr, *.mli)
- **unwrap** – unwrap output file (*.unw)
- **flag** – unwapping flag file (*.flag)
- **width** – interferogram width
- **lines** – number of interferogram lines
- **corr_thr** – threshold for correlation in the unwrapping mask (default=.3)
- **pwr_thr** – intensity threshold for phase unwrapping neutrons, multiples of average (default = 6.)
- **r_init** – range seed location in the interferogram

- **az_init** – azimuth seed location in the interferogram
- **r1** – starting range sample offset to unwrap
- **r2** – ending range sample offset to unwrap
- **l1** – starting line offset to unwrap
- **l2** – ending line offset to unwrap
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**UNWRAP_PAR**(*interf_par*, *interf*, *cc*, *pwr*, *unwrap*, *flag*, *corr_thr='-'*, *pwr_thr='-'*, *r_init='-'*, *az_init='-'*, *r1='-'*, *r2='-'*, *l1='-'*, *l2='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **interf_par** – interferogram parameter file *.off
- **interf** – interferogram filename (*.int, *.flt)
- **cc** – correlation filename (*.cc)
- **pwr** – intensity image (*.pwr, *.mli)
- **unwrap** – unwrap output file (*.unw)
- **flag** – unwapping flag file (*.flag)
- **corr_thr** – threshold for correlation in the unwrapping mask (default=.3)
- **pwr_thr** – intensity threshold for phase unwrapping neutrons, multiples of average (default = 6.)
- **r_init** – range seed location in the interferogram
- **az_init** – azimuth seed location in the interferogram
- **r1** – starting range sample offset to unwrap
- **r2** – ending range sample offset to unwrap
- **l1** – starting line offset to unwrap
- **l2** – ending line offset to unwrap
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**adapt_filt**(*int*, *sm*, *width*, *low_SNR_thr='-'*, *filt_width='-'*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Adaptive bandpass filtering of interferograms
Copyright 2016, Gamma Remote Sensing, v3.5 clw 17-Feb-2016

Parameters

- **int** – (input) complex interferogram image filename

- **sm** – (output) smoothed interferogram filename
- **width** – number of samples/row
- **low_SNR_thr** – low SNR threshold (default = .25);
- **filt_width** – filter width in pixels (default = 1.0)
- **xmin** – offset to starting range pixel(default = 0)
- **xmax** – offset last range pixel (default = width-1)
- **ymin** – offset to starting azimuth row (default = 0)
- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**adf**(*interf*, *sm*, *cc*, *width*, *alpha='-'*, *nfft='-'*, *cc_win='-'*, *step='-'*, *loff='-'*, *nlines='-'*, *wfrac='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Adaptive spectral filtering for complex interferograms
Copyright 2017, Gamma Remote Sensing, v3.6 12-Oct-2017 clw

Parameters

- **interf** – (input) interferogram (fcomplex)
- **sm** – (output) filtered interferogram (fcomplex)
- **cc** – (output) coherence derived from filtered interferogram (float)
- **width** – number of samples/line
- **alpha** – exponent for non-linear filtering (enter - for default: 0.40)
- **nfft** – filtering FFT window size, 2**N, 8 –> 512, (enter - for default: 32)
- **cc_win** – coherence parameter estimation window size odd, max: 15 (enter - for default: 5)
- **step** – processing step (enter - for default: nfft/8)
- **loff** – offset to starting line to process (enter - for default: 0)
- **nlines** – number of lines to process (enter - for default: to end of file)
- **wfrac** – minimum fraction of points required to be non-zero in the filter window (enter - for default: 0.200)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**af_SLC**(*SLC_par*, *SLC*, *rwin='-'*, *azwin='-'*, *dr='-'*, *daz='-'*, *thres='-'*, *a1_flg='-'*, *b0_flg='-'*, *offsets='-'*, *n_ovr='-'*, *roff='-'*, *azoff='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Focus testing for SLC data using autofocus estimation of effective velocity
Copyright 2016, Gamma Remote Sensing, v1.4 16-Feb-2016 clw/uw

Parameters

- **SLC_par** – (input) ISP SLC image parameter file
- **SLC** – (input) single-look complex image
- **rwin** – range window size (enter - for default: 1024)
- **azwin** – azimuth window size (enter - for default: 4096)
- **dr** – range sample increment (enter - for default: 1024, enter 0 for single patch)
- **daz** – azimuth line increment (enter - for default: 8192, enter 0 for single patch)
- **thres** – offset estimation SNR threshold (enter - for default: 10.000)
- **a1_flg** –

  fit a1 for first derivative of the effective velocity w.r.t.range

  - 0: no (default)
  - 1: yes
- **b0_flg** –

  fit b0 for first derivative of the effective velocity w.r.t. along-track time

  - 0: no (default)
  - 1: yes
- **offsets** – (output) range and azimuth offsets and SNR data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (1,2,4: enter - for default: 1)
- **roff** – range offset for single patch center
- **azoff** – azimuth offset for single patch center
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ave_image**(*im_list*, *width*, *ave*, *start='-'*, *nlines='-'*, *pixav_x='-'*, *pixav_y='-'*, *zflag='-'*, *nmin='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate average of a stack of images (float format)
Copyright 2015, Gamma Remote Sensing, v1.9 20-Nov-2015 clw

Parameters

- **im_list** – (input) text file with names of co-registered images in column 1 (float)
- **width** – number of samples/line
- **ave** – (output) average of input image data files (float)
- **start** – starting line (default: 1)
- **nlines** – number of lines to process (enter - for default: entire file)
- **pixav_x** – number of pixels to average in width (default: 1)
- **pixav_y** – number of pixels to average in height (default: 1)
- **zflag** –

  zero flag:

  - 0: interpret 0.0 as missing data value (default)

  - 1: interpret 0.0 as valid data

- **nmin** – minimum number of images required to calculate the average if zflag = 0 (default: 3/4*nfiles)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**az_integrate**(*data*, *width*, *azi*, *cflg*, *scale='-'*, *lz='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate azimuth integral of float data (unwrapped phase or azimuth offsets)
Copyright 2012, Gamma Remote Sensing, v1.2 6-Feb-2012

Parameters

- **data** – (input) input data (example: SBI dtrapped phase) (float)
- **width** – (input) number of range samples/line
- **azi** – (output) input data integrated along azimuth (float)
- **cflg** –

  integration constant flag:

  - 0: set azimuth integral value to 0.0 at specified line
  - 1: set average of the azimuth integral to 0.0

- **scale** – scale factor to apply to the data (enter - for default, default: 1.0)
- **lz** – line offset where the azimuth integral is set to 0.0 (cflg = 0, enter - for default, default: 0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**az_spec_SLC**(*SLC*, *SLC_par*, *spectrum*, *roff='-'*, *namb='-'*, *pltflg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Doppler centroid estimate from SLC images
Copyright 2016, Gamma Remote Sensing, v2.9 clw 15-Feb-2016

Parameters

- **SLC** – (input) SAR image data file (fcomplex or scomplex format)
- **SLC_par** – (input) ISP SLC image parameter file
- **spectrum** – (output) Doppler spectrum (text format)
- **roff** – range sample offset to center of estimation window (enter - for default=center_swath)
- **namb** – number of multiples of the PRF to add to the estimated centroid (default=0)
- **pltflg** –

  azimuth spectrum plotting flag:

  - 0: none (default)

- 1: output plot in PNG format

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**base_copy**(*SLC1_par*, *baseline_1*, *SLC2_par*, *baseline_2*, *time_rev='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate baseline file for a subsection of a reference SLC
Copyright 2003, Gamma Remote Sensing, v1.1 6-Jan-2003 ts/clw/uw

Parameters

- **SLC1_par** – (input) ISP image parameter file of the reference SLC
- **baseline_1** – (input) baseline file derived using the reference SLC geometry
- **SLC2_par** – (input) ISP image parameter file corresponding to the subsecton of the reference SLC
- **baseline_2** – (output) baseline file derived using the geometry and timing of the SLC subsection
- **time_rev** – SLC image normal=1, time-reversed = -1 (default=1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**base_est_fft**(*interf*, *SLC1_par*, *OFF_par*, *baseline*, *nazfft='-'*, *r_samp='-'*, *az_line='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Estimate baseline from interferogram fringe spectrum
Copyright 2016, Gamma Remote Sensing, v2.1 clw/uw 20-Feb-2016

Parameters

- **interf** – (input) multi-look interferogram with range phase
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **baseline** – (output) baseline file
- **nazfft** – size of azimuth FFT (lines read from file, 2**N) (default: 512)
- **r_samp** – range pixel offset to center of the FFT window (default: center)
- **az_line** – line offset from start of the interf. for the FFT window (default=center)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**base_init**(*SLC1_par*, *SLC2_par*, *OFF_par*, *interf*, *baseline*, *mflag='-'*, *nrfft='-'*, *nazfft='-'*, *r_samp='-'*, *az_line='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Estimate initial baseline using orbit state vectors, offsets, and interferogram phase
Copyright 2016, Gamma Remote Sensing, v2.5 clw/uw 19-Feb-2016

Parameters

- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file (enter - for none)
- **interf** – (input) unflattened interferogram (enter - for none)
- **baseline** – (output) baseline parameter file
- **mflag** – baseline estimation method flag (enter - for default)
- **mflag** –

  b_para b_perp input

  - 0: orbits orbits p1,p2 (default)
  - 1: offsets offsets p1,p2,off
  - 2: orbits fft p1,p2,off,int
  - 3: offsets fft p1,p2,off,int
  - 4: fft fft p1,off,int
- **nrfft** – size of range FFT (512, 1024,...) (enter - for default determined from image width)
- **nazfft** – size of azimuth FFT (512, 1024,...) (enter - for default determined from image azimuth lines)
- **r_samp** – range pixel offset to center of the FFT window (enter - for default, default: range center)
- **az_line** –

  line offset from start of the interf. for the FFT window (enter - for default, default: azimuth center)

  - NOTE: Not all input data files are required for the different methods enter - for files that are not provided
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**base_ls**(*SLC_par*, *OFF_par*, *gcp_ph*, *baseline*, *ph_flag='-'*, *bc_flag='-'*, *bn_flag='-'*, *bcdot_flag='-'*, *bndot_flag='-'*, *bperp_min='-'*, *SLC2R_par='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Least squares baseline estimation using terrain heights
Copyright 2005, Gamma Remote Sensing, v2.2 5-Sep-2005 clw/uw

Parameters

- **SLC_par** – (input) ISP parameter file of the reference SLC
- **OFF_par** – (input) ISP interferogram/offset parameter file
- **gcp_ph** – (input) ground control point heights + extracted unwrapped phase (text format)
- **baseline** – (input) baseline parameter file

- **ph_flag** – restore range phase ramp (default=0: do not restore 1: restore)
- **bc_flag** – cross-track baseline component estimate (0:orbit derived 1:estimate from data, default=1)
- **bn_flag** – normal baseline component estimate (0:orbit derived 1:estimate from data, default=1)
- **bcdot_flag** – cross-track baseline rate estimate (0:orbit derived 1:estimate from data, default=1)
- **bndot_flag** – normal baseline rate estimate (0:orbit derived 1:estimate from data, default=0)
- **bperp_min** – minimum perpendicular baseline required for L.S estimation (m, default= 10.0)
- **SLC2R_par** – (input) parameter file of resampled SLC, required if SLC-2 frequency differs from SLC-1
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**base_orbit**(*SLC1_par*, *SLC2_par*, *baseline*, *logpath=None*, *outdir=None*, *shellscript=None*)

Estimate baseline from orbit state vectors
Copyright 2015, Gamma Remote Sensing, v4.2 clw 18-Apr-2018

Parameters

- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **baseline** – (output) baseline file (text format, enter - for none)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**base_perp**(*baseline*, *SLC1_par*, *OFF_par*, *time_rev='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate baseline components perpendicular and parallel to look vector
Copyright 2005, Gamma Remote Sensing, v3.5 10-May-2005 clw/uw

Parameters

- **baseline** – (input) baseline file (text)
- **SLC1_par** – (input) ISP parameter file of SLC-1 (reference SLC)
- **OFF_par** – (input) ISP interferogram/offset parameter file
- **time_rev** – SLC image normal=1 (default), image time-reversed = -1
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**bpf**(*data_in*, *data_out*, *width*, *fc_x*, *bw_x*, *fc_y*, *bw_y*, *roff='-'*, *azoff='-'*, *nr='-'*, *naz='-'*, *data_type='-'*, *f_mode='-'*, *beta='-'*, *fir_len='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Interferometric SAR Processor (ISP): Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/bpf.c
Copyright 2016, Gamma Remote Sensing, v1.7 clw 3-Mar-2016
Bandpass filter for 2-dimensional complex image data (FCOMPLEX or SCOMPLEX format)

Parameters

- **data_in** – (input) input data file (fcomplex, scomplex, float)
- **data_out** – (output) output data file (fcomplex, scomplex, float)
- **width** – number of samples/line
- **fc_x** – normalized x-coord. (across) filter center frequency (range: -0.5 –> 0.5)
- **bw_x** – normalized x-coord. bandwidth (range: 0 –> 1.0)
- **fc_y** – normalized y-coord. (down) filter center frequency (range: -0.5 –> 0.5)
- **bw_y** – normalized y-coord. bandwidth (range: 0 –> 1.0)
- **roff** – offset to starting range to filter (default: 0)
- **azoff** – offset to starting azimuth to filter (default: 0)
- **nr** – number of range pixels to filter (default - : width - roff)
- **naz** – number of azimuth lines to filter (default - : nlines - azoff)
- **data_type** – data type (default 0:fcomplex, 1:scomplex, 2:float)
- **f_mode** – fill mode (default 0:force filtered value to 0.0 for input value 0.0, 1:no forcing)
- **beta** – Kaiser window beta parameter (default - : 1.000)
- **fir_len** – finite impulse reponse filter length (default - : 128)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**bpf_ssi**(*SLC*, *SLC_par*, *SLC_flow*, *SLC_flow_par*, *SLC_fhigh*, *SLC_fhigh_par*, *rbs='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **SLC** – (input) SLC (FCOMPLEX or SCOMPLEX, SLC should not be resampled)
- **SLC_par** – (input) SLC parameter file
- **SLC_flow** – (output) low frequency band filtered SLC (FCOMPLEX or SCOMPLEX)
- **SLC_flow_par** – (output) low frequency band filtered SLC parameter file
- **SLC_fhigh** – (output) high frequency band filtered SLC (FCOMPLEX or SCOMPLEX)
- **SLC_fhigh_par** – (output) high frequency band filtered SLC parameter file (FCOMPLEX or SCOMPLEX)
- **rbs** –

relative range spectrum band separation (default = 0.6666 –> lowest and highest third of processing bandwidth)
indicate - for the output files to only calculate filtering parameters

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**bridge**(*int*, *flag*, *unw*, *bridge*, *width*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Phase unwrap new regions with bridges to regions already unwrapped
Copyright 2010, Gamma Remote Sensing, v1.5 clw 4-Nov-2010

Parameters

- **int** – (input) interferogram (fcomplex)
- **flag** – (input) unwrapping flag file
- **unw** – (input/output) unwrapped phase (float)
- **bridge** – (input) bridge data file (text format)
- **width** – number of samples/row
- **xmin** – starting range pixel offset to unwrap (default = 0)
- **xmax** – last range pixel offset to unwrap (default=width-1)
- **ymin** – starting azimuth row offset to unwrap, relative to start (default = 0)
- **ymax** – last azimuth row offset to unwrap, relative to start (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**cc_wave**(*interf*, *MLI_1*, *MLI_2*, *cc*, *width*, *bx='-'*, *by='-'*, *wflg='-'*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Estimate interferometric coherence
Copyright 2017, Gamma Remote Sensing, v6.0 24-Oct-2017 clw/uw

Parameters

- **interf** – (input) normalized complex interferogram
- **MLI_1** – (input) intensity image of the first scene (float) (enter - for none)
- **MLI_2** – (input) intensity image of the second scene (float) (enter - for none)
- **cc** – (output) estimated degree of coherence filename
- **width** – number of samples/line
- **bx** – estimation window size in columns (enter - for default:5.0)
- **by** – estimation window size in lines (enter - for default:5.0)
- **wflg** –

  estimation window (enter - for default):

  - 0: rectangular (default)
  - 1: triangular
  - 2: Gaussian
  - 3: normalized vector sum with rectangular window
  - NOTE: This estimator does not use the MLI data, even when specified
- **xmin** – starting range pixel offset (enter - for default: 0)
- **xmax** – last range pixel offset (enter - for default: width - 1)
- **ymin** – starting azimuth row offset, relative to start (enter - for default: 0)
- **ymax** – last azimuth row offset, relative to start (enter - for default: nlines - 1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**clear_flag**(*flag*, *width*, *flag_bits*, *xmin*, *xmax*, *ymin*, *ymax*, *logpath=None*, *outdir=None*, *shellscript=None*)

Clear phase unwrapping flag bits
Copyright 2005, Gamma Remote Sensing, v1.6 clw 17-Oct-2005

Parameters

- **flag** – (input)phase unwrapping flag filename
- **width** – number of samples/row
- **flag_bits** –

  byte with value of flag(s) to be cleared:
  Charges = 3 Guides = 4 Low SNR = 8 Visited = 16 BRANCH PT. = 32 Cuts = 64 Lawn = 128
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row offset, relative to start (default = 0)
- **ymax** – last azimuth row offset, relative to start (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**corr_flag**(*corr*, *flag*, *width*, *corr_thr*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *border='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Low correlation region detection for phase unwrapping
Copyright 2005, Gamma Remote Sensing, v2.4 1-Mar-2005 clw/uw

Parameters

- **corr** – (input)interferometric correlation file
- **flag** – (input/output) phase unwrapping flag filename
- **width** – number of samples/row

- **corr_thr** – corrrelation threshold (0 –> 1.0)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row offset, relative to start (default = 0)
- **ymax** – last azimuth row offset, relative to start (default = nlines-1)
- **border** – effective range of low coherence pixels to set low coherence flag (default=2)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**create_offset**(*SLC1_par*, *SLC2_par*, *OFF_par*, *algorithm='-'*, *rlks='-'*, *azlks='-'*, *iflg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Create and update ISP offset and interferogram parameter files
Copyright 2015 Gamma Remote Sensing v5.3 clw/uw 10-Nov-2015

Parameters

- **SLC1_par** – (input) SLC-1/MLI-1 ISP image parameter filename (reference)
- **SLC2_par** – (input) SLC-2/MLI-2 ISP image parameter filename
- **OFF_par** – (input/output) ISP offset/interferogram parameter file
- **algorithm** –

  offset estimation algorithm

  - 1: intensity cross-correlation (default)
  - 2: fringe visibility
- **rlks** – number of interferogram range looks (enter - for default: 1)
- **azlks** – number of interferogram azimuth looks (enter - for default: 1)
- **iflg** –

  interactive mode flag (enter - for default)

  - 0: non-interactive
  - 1: interactive (default)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**dcomp_sirc**(*infile*, *outfile*, *samples*, *loff='-'*, *nlines='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract SIR-C SLC compressed single-pol data
Copyright 2009, Gamma Remote Sensing, v1.4 16-Oct-2009 clw

Parameters

- **infile** – (input) SIR-C single-pol SLC compressed data

- **outfile** – (output) complex floating point data
- **samples** – number of polarimetric samples per input line (4 bytes/sample)
- **loff** – offset to starting line (default: 0)
- **nlines** – number of lines to copy(default: entire file, 0 = entire file)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**dcomp_sirc_quad**(*infile*, *outfile*, *samples*, *parameter*, *loff='-'*, *nlines='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract SIR-C MLC or SLC compressed quad-pol data
Copyright 2009, Gamma Remote Sensing, v1.4 16-Oct-2009 uw/clw

Parameters

- **infile** – (input) SIR-C SLC or MLC quad-pol compressed data
- **outfile** – (output) complex floating point data
- **samples** – number of polarimetric samples per input line (10 bytes/sample)
- **parameter** –

  polarimetric parameter to extract from SLC or MLC product:

  - 0: SLC total power
  - 1: SLC-HH
  - 2: SLC-HV
  - 3: SLC-VH
  - 4: SLC-VV
  - 5: MLC total power
  - 6: MLC-HVHV*
  - 7: MLC-VVVV*
  - 8: MLC-HHHH*
  - 9: MLC-HHHV*
  - 10: MLC-HHVV*
  - 11: MLC-HVVV*
- **loff** – offset to starting line (default: 0)
- **nlines** – number of lines to copy(default: entire file, 0 = entire file)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**error_stat**(*d1*, *d2*, *width*, *dtype*, *roff*, *loff*, *nr*, *nl*, *report*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate statistics for two data files and their difference (FLOAT or FCOMPLEX)

Copyright 2017, Gamma Remote Sensing, v1.2 clw 7-Jan-2016

Parameters

- **d1** – (input) data file 1
- **d2** – (input) data file 2
- **width** – image line width (samples/line)
- **dtype** –

  data type for d1 and d2:

  - 0: FLOAT
  - 1: FCOMPLEX
- **roff** – sample offset to region start (enter - for default: 0)
- **loff** – line offset to region start (enter - for default: 0)
- **nr** – region width (samples, enter - for default: width - roff)
- **nl** – number of lines in the region (enter - for default: data_lines - loff)
- **report** –

  output text file (keyword:value format)
  keywords: data_1, data_2, d1_mean, d2_mean, d1_stddev, d2_stddev, root_mean_square_error, normalized_mean_square_error, cross_correlation_coefficient, cross_correlation_angle, total_samples, non_zero_samples, valid_fraction
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**fill_gaps**(*data_in*, *width*, *data_out*, *method='-'*, *max_dist='-'*, *bp_flag='-'*, *win='-'*, *ds_method='-'*, *ds_size='-'*, *ds_data='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Fill gaps in 2D raster file
Copyright 2017, Gamma Remote Sensing, v1.6 6-Apr-2017 cm

Parameters

- **data_in** – (input) input data file (float)
- **width** – width of input data
- **data_out** – (output) output data file (float)
- **method** –

  method flag (enter - for default: 1)

  - 0: Laplace interpolation and linear extrapolation - least squares solution
  - 1: Laplace interpolation and linear extrapolation - smaller system of linear equations than in method #0 in case of few missing values - least squares solution (default)
  - 2: Laplace interpolation and linear extrapolation - solves a direct linear system of equations for the missing values (not a least squares solution)
  - 3: biharmonic interpolation - implementation similar to method #1 - least squares solution

  - 4: spring analogy: assumes springs (with a nominal length of zero) connect each node with every neighbor - least squares solution
  - 5: average of the 8 nearest neighbors - this method solves a direct linear system for the missing values (not a least squares solution) hints: small gaps: use method #0, #1 or #3 - large gaps: use method #2, #4 or #5 - most demanding: method #3
- **max_dist** – maximum interpolation / extrapolation distance in pixels (enter - or 0 for default: unlimited)
- **bp_flag** –

  perform block processing (enter - for default: 0)
  - 0: no block processing (default)
  - 1: block processing (faster, avoid overflow, however might be slightly less accurate) when block processing is selected, a two-step process is carried out: 1: solving the downsampled array (coarse processing), 2: block processing
- **win** – block size (pixels, 10 < win < 1000, enter - for default: 100)
- **ds_method** –

  method flag (0 - 5, same choices as for [method] option) (enter - for default: same as [method])
  hint: for an input containing large gaps, method #2, #4 or #5 may yield more appropriate results.
- **ds_size** – maximum size of downsampled data (for both width and height) (pixels, ds_size > 10, enter - for default: 400)
- **ds_data** – (output) write intermediate data after solving the downsampled array (float)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**fspf**(*data_in*, *data_out*, *width*, *dtype='-'*, *r_max='-'*, *spf_type='-'*, *MLI_par='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/fspf.c
Copyright 2014, Gamma Remote Sensing, v1.2 28-May-2014 of/clw/uw
Fast spatial filter for 2D data

Parameters

- **data_in** – (input) input image data
- **data_out** – (output) spatially filtered image data
- **width** – number of samples/row
- **dtype** –

  data type (enter - for default):
  - 0: FCOMPLEX
  - 1: SCOMPLEX
  - 2: FLOAT (default)
- **r_max** – maximum filter radius (range samples) (enter - for default: 64)
- **spf_type** –

  spatial filter type (enter - for default):

- 0: uniform average (default for fcomplex and scomplex)
- 1: triangular weighted average: 1 - (r/r_max)
- 2: quadratic weighted average: 1 - (r/r_max)**2
- 3: Gaussian weighted average: exp(-2.*(r**2/r_max**2))
- 4: linear least-squares (default for float data)

- **MLI_par** – MLI or SLC parameter file with the same number of looks as the input image, required for GPRI data
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**gcp_phase**(*unw*, *OFF_par*, *gcp*, *gcp_ph*, *win_sz='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract unwrapped phase at GCP locations
Copyright 2006, Gamma Remote Sensing, v1.5 8-Mar-2006 clw

Parameters

- **unw** – (input) unwrapped interferometric phase
- **OFF_par** – (input) ISP interferogram/offset parameter file
- **gcp** – (input) ground control point data (text format)
- **gcp_ph** – (output) ground control point data + extracted unwrapped phase (text)
- **win_sz** – window size for averaging phase for each GCP, must be odd (default: 1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**get_GAMMA_RASTER**(*mode*, *logpath=None*, *outdir=None*, *shellscript=None*)

Script to determine the default extension for raster images or the operating system type v1.1 clw/uw 11-Nov-2016

Parameters

- **mode** –

  Specify the script string output:

  - 0: raster file extension (ras, bmp, or tif)
  - 1: OS type: Linux, MINGW64_NT-10.0, CYGWIN_NT-10.0, darwin...
  - NOTE: The default raster format on Linux systems is SUN_RASTER (*.ras), for all other operating systems it is BMP (*.bmp). SUN_RASTER and BMP images are limited in size to 32767 x 32767. TIFF files do not have this limitation. To set the default image raster format for Gamma programs, set the environment variable GAMMA_RASTER as follows: bash: export GAMMA_RASTER=SUN_RASTER #extension: ras export GAMMA_RASTER=BMP #extension: bmp export GAMMA_RASTER=TIFF #extension: tif csh,tcsh: setenv GAMMA_RASTER SUN_RASTER #extension: ras setenv GAMMA_RASTER BMP #extension: bmp setenv GAMMA_RASTER

TIFF #extension: tif Environment variables can be set either in processing scripts, or in the shell initialization file (e.g. .bashrc) Programs in the Gamma software that generate raster image files query the value of GAMMA_RASTER if it has been defined. This script can be called from within another script to determine the default raster image format or OS type: bash: $ext=`get_GAMMA_RASTER 0` csh,tcsh: set ext=`get_GAMMA_RASTER 0` The variable $ext can then be used to specify the format of the output raster file by using it to construct the output file name: bash: $my_raster=$my_name"."$ext csh/tcsh: set my_raster=$my_name"."$ext OS: Linux GAMMA_RASTER: Undefined variable.

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**grasses**(*int*, *flag*, *unw*, *width*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *xinit='-'*, *yinit='-'*, *init_ph='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Phase unwrapping by region growing
Copyright 2005, Gamma Remote Sensing, v4.2 1-Mar-2005 clw/uw

Parameters

- **int** – (input) interferogram filename
- **flag** – (input) unwrapping flag filename
- **unw** – (output) unwrapped phase filename
- **width** – number of samples/row
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default=width-1)
- **ymin** – starting azimuth row offset, relative to start (default = 0)
- **ymax** – last azimuth row offset, relative to start (default = nlines-1)
- **xinit** – starting range pixel for unwrapping (default = width/2)
- **yinit** – starting row to unwrap (default = height/2)
- **init_ph** – flag to set phase at starting point to 0.0 (default 0: not set to 0.0, 1: set to 0.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**hgt_map**(*unw*, *SLC_par*, *OFF_par*, *baseline*, *hgt*, *gr*, *ph_flag='-'*, *loff='-'*, *nlines='-'*, *SLC2R_par='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Interferometric height/ground range estimation vs. slant range
Copyright 2005, Gamma Remote Sensing, v5.1 clw/uw 9-Sep-2005

Parameters

- **unw** – (input) unwrapped interferometric phase
- **SLC_par** – (input) ISP parameter file for the reference SLC

- **OFF_par** – (input) ISP offset/interferogram processing parameters
- **baseline** – (input) baseline parameter file
- **hgt** – (output) height file (in slant range geometry) relative to the WGS-84 ellipsoid
- **gr** – (output) cross-track ground ranges on the WGS-84 ellipsoid (in slant range geometry)
- **ph_flag** – restore phase slope flag (0:no phase change default=1:add back phase ramp)
- **loff** – offset to starting line (default = 0)
- **nlines** – number of lines to calculate (enter - for default: to end of file)
- **SLC2R_par** – (input) parameter file of resampled SLC, required if SLC-2 frequency differs from SLC-1
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**image_stat**(*image*, *width*, *roff*, *loff*, *nr*, *nl*, *report*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate mean, standard deviation and number of non-zero values for a rectangular image region (float format)
Copyright 2016, Gamma Remote Sensing, v1.3 3-Nov-2016

Parameters

- **image** – (input) image data file (float)
- **width** – image line width (samples/line)
- **roff** – sample offset to region start (enter - for default: 0)
- **loff** – line offset to region start (enter - for default: 0)
- **nr** – region width (samples, enter - for default: width - roff)
- **nl** – number of lines in the region (enter - for default: image_lines - loff)
- **report** –

  output text file (keyword:value format)
  keywords: file, mean, stdev, total_samples, non_zero_samples, fraction_valid)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**init_offset**(*SLC_1*, *SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *rlks='-'*, *azlks='-'*, *rpos='-'*, *azpos='-'*, *offr='-'*, *offaz='-'*, *thres='-'*, *rwin='-'*, *azwin='-'*, *cflag='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Determine initial offset between SLC images using correlation of image intensity
Copyright 2016, Gamma Remote Sensing, v3.1 clw 12-Apr-2016

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)

- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **rlks** – number of range looks (default: 1)
- **azlks** – number of azimuth looks (default: 1)
- **rpos** – center of patch in range (samples) (enter - for default: image center)
- **azpos** – center of patch in azimuth (lines) (enter - for default: image center)
- **offr** – initial range offset (samples) (enter - for default: 0)
- **offaz** – initial azimuth offset (lines) (enter - for default: 0)
- **thres** – cross-correlation threshold (enter - for default: 0.150)
- **rwin** – range window size (default: 512)
- **azwin** – azimuth window size (default: 512)
- **cflag** –

  copy offsets to the range and azimuth offset polynomials in the OFF_par

  - 0: do not copy
  - 1: copy constant range and azimuth offset (default)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**init_offset_orbit**(*SLC1_par*, *SLC2_par*, *OFF_par*, *rpos='-'*, *azpos='-'*, *cflag='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Initial SLC image offset estimation from orbit state-vectors and image parameters
Copyright 2016, Gamma Remote Sensing, v1.7 21-Apr-2016 clw/uw

Parameters

- **SLC1_par** – (input) SLC-1 parameter file
- **SLC2_par** – (input) SLC-2 parameter file
- **OFF_par** – (input/output) ISP/offset parameter file
- **rpos** – range position for offset estimation (enter - for default: center of SLC-1)
- **azpos** – azimuth position for offset estimation (enter - for default: center of SLC-1)
- **cflag** –

  copy offsets to the range and azimuth offset polynomials in the OFF_par

  - 0: do not copy
  - 1: copy constant range and azimuth offset (default)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**interf_SLC**(*SLC_1*, *SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *MLI_1*, *MLI_2*, *interf*, *nrlk='-'*, *nazlk='-'*, *loff='-'*, *nltot='-'*, *rfilt='-'*, *azfilt='-'*, *s_off='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Interferogram generation using a pair of SLC images
Copyright 2009, Gamma Remote Sensing, v4.10 clw/uw 27-Oct-2009

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **MLI_1** – (output) multi-look intensity image 1
- **MLI_2** – (output) multi-look intensity image 2
- **interf** – interferogram from SLC-1 and SLC-2
- **nrlk** – number of interferogram range looks (default: 2)
- **nazlk** – number of interferogram azimuth looks (default: 10)
- **loff** – offset to starting line of interferogram (relative to start of SLC-1) (default: 0)
- **nltot** – number of SLC lines to process (default: 0, to end of file)
- **rfilt** –

  range common band filtering flag

  - 0: OFF
  - 1: ON (default)
- **azfilt** –

  azimuth common band filtering flag

  - 0: OFF
  - 1: ON (default)
- **s_off** –

  offset to the nominal range spectral shift (frac. of range sampling freq.) (default: 0.0)

  - NOTE: enter - as filename to avoid creation of corresponding output file
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**interp_ad**(*data_in*, *data_out*, *width*, *r_max='-'*, *np_min='-'*, *np_max='-'*, *w_mode='-'*, *type='-'*, *cp_data='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Weighted interpolation of gaps in 2D data using an adaptive smoothing window
Copyright 2018, Gamma Remote Sensing, v2.1 13-Jun-2018 clw/uw

Parameters

- **data_in** – (input) data with gaps
- **data_out** – (output) data with gaps filled by interpolation
- **width** – number of samples/row
- **r_max** – maximum interpolation window radius (default(-): 16)
- **np_min** – minimum number of points used for the interpolation (default(-): 16)
- **np_max** – maximum number of points used for the interpolation (default(-): 16)
- **w_mode** –

  data weighting mode (enter - for default):

  - 0: constant
  - 1: 1 - (r/r_max)
  - 2: 1 - (r/r_max)**2 (default)
  - 3: exp(-2.*(r**2/r_max**2))
- **type** –

  input and output data type:

  - 0: FCOMPLEX
  - 1: SCOMPLEX
  - 2: FLOAT (default)
  - 3: INT
  - 4: SHORT
- **cp_data** –

  copy data flag:

  - 0: do not copy input data values to output
  - 1: copy input data values to output (default)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ionosphere_check**(*SLC*, *par*, *rwin='-'*, *azwin='-'*, *thresh='-'*, *rstep='-'*, *azstep='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **SLC** – (input) SLC image (e.g. 20070214.slc)
- **par** – (input) SLC parameter file (e.g. 20070214.slc.par)
- **rwin** – range window size used in offset estimation (default = 256)
- **azwin** – azimuth window size used in offset estimation (default = 256)
- **thresh** – threshold value used in offset estimation (default = 0.1)
- **rstep** – range step used in offset estimation (default = rwin/4)

- **azstep** – azimuth step used in offset estimation (default = azwin/4)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**make_tab**(*list*, *tab*, *definition*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **list** – (input) list file (ascii)
- **tab** – (output) table file (ascii)
- **definition** –

  definition used to generate line of output table
  (example 1: '$1.slc $1.slc.par') (example 2: '$1_$2.base $1_$2.off')
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**mask_data**(*data_in*, *width*, *data_out*, *mask*, *dtype='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Mask float or fcomplex data using an 8-bit SUN/BMP/TIFF format raster image
Copyright 2016, Gamma Remote Sensing, v1.4 10-Dec-2016 clw

Parameters

- **data_in** – (input) data file (FLOAT or FCOMPLEX format)
- **width** – width of input data file
- **data_out** – (output) data file, same data format as input
- **mask** –

  (input) mask file, SUN/BMP/TIFF raster format, 8-bits/pixel
  output data values are set to 0.0 at all locations where the mask is black (0,0,0) or dn = 0 * NOTE: mask file must have the same width as the input data file
- **dtype** –

  data format:

  - 0: FLOAT (default)
  - 1: FCOMPLEX
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**mcf**(*interf*, *wgt*, *mask*, *unw*, *width*, *tri_mode='-'*, *roff='-'*, *loff='-'*, *nr='-'*, *nlines='-'*, *npat_r='-'*, *npat_az='-'*, *ovrlap='-'*, *r_init='-'*, *az_init='-'*, *init_flag='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Phase unwrapping using Minimum Cost Flow (MCF) and triangulation
Copyright 2016, Gamma Remote Sensing, v2.2 clw/uw 30-Nov-2016

Parameters

- **interf** – (input) interferogram (*.int,*.flt)(fcomplex)
- **wgt** – (input) weight factors (0 -> 1.0) file (float)(enter - for uniform weight)
- **mask** – (input) validity mask (SUN/BMP/TIFF raster format, value 0 -> pixel not used)(enter - if no mask)
- **unw** – (output) unwrapped phase image (*.unw)(float)
- **width** – number of samples/row
- **tri_mode** –

  triangulation mode

  - 0: filled triangular mesh (default)
  - 1: Delaunay triangulation
- **roff** – offset to starting range of section to unwrap (default: 0)
- **loff** – offset to starting line of section to unwrap (default: 0)
- **nr** – number of range samples of section to unwrap (default(-): width - roff)
- **nlines** – number of lines of section to unwrap (default(-): total number of lines - loff)
- **npat_r** – number of patches in range
- **npat_az** – number of patches in azimuth
- **ovrlap** – overlap between patches in pixels (overlap >= 7, default(-): 512)
- **r_init** – phase reference point range offset (default(-): roff)
- **az_init** – phase reference point azimuth offset (default(-): loff)
- **init_flag** –

  flag to set phase at reference point

  - 0: use initial point phase value (default)
  - 1: set phase to 0.0 at initial point
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**mk_ptarg**(*RSLC_tab*, *cal_dir*, *r_samp*, *az_samp*, *osf='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/mk_ptarg
Copyright 2012, Gamma Remote Sensing, v1.5 24-Apr-2012 clw
Perform point target analysis on a stack of coregistered SLCs

Parameters

- **RSLC_tab** –

  (input) two column list of coregistered SLC filenames and SLC parameter filenames (including paths) (ascii)

  1. SLC filename (includes path)
  2. SLC parameter filename (includes path)

- **cal_dir** – directory for output calibration results
- **r_samp** – (input) calibration target range sample number
- **az_samp** – (input) calibration target azimuth line number
- **osf** –

  SLC over-sampling factor 2, 4, 8, 16, 32, 64 (default: 16)

-s scale	(option) set image display scale factor (default: 0.3)
-e exp	(option) set image display exponent (default: 0.5)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**mk_ptarg_cal**(*CR_tab*, *SLC*, *SLC_par*, *cal_dir*, *sigma*, *c_rpos*, *c_azpos*, *osf='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

/usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/mk_ptarg_cal
Copyright 2016, Gamma Remote Sensing, v1.8 19-Feb-2016 clw
Perform point target analysis and calibration factor evaluation for a set of point targers

Parameters

- **CR_tab** –

  (input) 3 column list of row and sample number of corner reflectors

  1. Corner reflector id
  2. SLC column (includes path)
  3. SLC row (includes path)

- **SLC** – SLC image
- **SLC_par** – SLC_parameter file
- **cal_dir** – directory for output calibration results
- **sigma** – Radar cross-section of the corner reflectors
- **c_rpos** – range sample number of the center of the region used to estimate region
- **c_azpos** – azimuth line of the center of the region used to estimate clutter
- **osf** –

  SLC over-sampling factor 2, 4, 8, 16, 32, 64 (default: 16)

-s scale	(option) set image display scale factor (default: 0.2)

-e exp	(option) set image display exponent (default: 0.5)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**mk_tab3**(*dir*, *ext_1*, *ext_2*, *ext_3*, *tab*, *logpath=None*, *outdir=None*, *shellscript=None*)

Copyright 2014, Gamma Remote Sensing, v1.0 27-Jun-2014 clw
Generate SLC_tab, MLI_tab, or RAW_list for processing

Parameters

- **dir** – (input) directory including paths that contain the data files
- **ext_1** – (input) pattern to select data files (examples: slc, raw. . .), (enter - for all files in the directory)
- **ext_2** – (input) pattern to select parameter files that match the data (enter - for none, examples: slc.par, raw_par, raw.par)
- **ext_3** – (input) pattern to select parameter files that match the data (enter - for none, examples: ppar)
- **tab** –

  (output) list of data filenames and associated parameter files (including paths) (text)
  - NOTE: The current directory is denoted using .
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**mosaic_WB**(*data_tab*, *dtype*, *data_out*, *data_par_out*, *sc_flg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP: Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/mosaic_WB.c
Copyright 2018, Gamma Remote Sensing, v1.3 26-Apr-2018 clw/cm
Mosaic Wide-Beam ScanSAR data processed by the MSP

Parameters

- **data_tab** – (input) 2 column list of data and ISP image parameter files for the beams in the mosaic (text)
- **dtype** –

  (input) input data type:
  - 0: FLOAT
  - 1: FCOMPLEX
- **data_out** – (output) output image mosaic
- **data_par_out** – (output) ISP image parameter file for output image mosaic
- **sc_flg** –

intensity scaling flag:

- 0: do not scale different beam data values
- 1: use overlap regions to scale beam intensities (default)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**multi_S1_TOPS**(*SLC_tab*, *MLI*, *MLI_par*, *rlks*, *azlks*, *wflg='-'*, *SLCR_tab='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate MLI mosaic from Sentinel-1 TOPS SLC burst data (FCOMPLEX and SCOMPLEX)
Copyright 2018, Gamma Remote Sensing v3.4 26-Apr-2018 awi/clw/uw/cm

Parameters

- **SLC_tab** – (input) 3 column list of SLC, SLC_par, Sentinel-1 TOPS_par, rows sorted in the order IW1, IW2, IW3
- **MLI** – (output) multi-look intensity image
- **MLI_par** – (output) MLI image parameter file
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **wflg** –

  burst window calculation flag:

  - 0: use existing burst window parameters if they exist, otherwise calculate burst window parameters (default)
  - 1: calculate burst window parameters from burst parameters and the number of range and azimuth looks

- **SLCR_tab** –

  (input) SLC_tab of the reference scene, 3 column list of SLC, SLC_par, TOPS_par sorted in the order IW1, IW2, IW3

  - NOTE: When generating an MLI mosaic of a resampled SLC, the SLC_tab of the reference scene is required

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**multi_SLC_WSS**(*SLC*, *SLC_par*, *MLI*, *MLI_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate multi-look intensity image (MLI) from a ASAR Wide-Swath SLC
Copyright 2008, Gamma Remote Sensing v1.2 08-Jan-2008 clw/awi

Parameters

- **SLC** – (input) ASAR Wide-Swath SLC image
- **SLC_par** – (input) ASAR Wide-Swath SLC image parameter file
- **MLI** – (output) multi-look intensity image

- **MLI_par** – (output) MLI image parameter file
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**multi_cpx**(*data_in*, *OFF_par_in*, *data_out*, *OFF_par_out*, *rlks='-'*, *azlks='-'*, *loff='-'*, *nlines='-'*, *roff='-'*, *nsamp='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate multi-look averaged or interpolated 2D image (fcomplex data)
Copyright 2013, Gamma Remote Sensing, v2.5 28-Mar-2013 clw/uw

Parameters

- **data_in** – (input) input fcomplex image file
- **OFF_par_in** – (input) offset parameter file for input image
- **data_out** – (output) output multi-look or interpolated fcomplex data file
- **OFF_par_out** – (input/output) offset parameter file for output, if already exists, then used as input
- **rlks** – number of range looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **azlks** – number azimuth looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **loff** – line offset to starting line (default: 0)
- **nlines** – number of lines (default: 0, to end of file)
- **roff** – offset to starting range sample (default:0)
- **nsamp** – number of range samples to extract (default: 0, to end of line)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**multi_look**(*SLC*, *SLC_par*, *MLI*, *MLI_par*, *rlks*, *azlks*, *loff='-'*, *nlines='-'*, *scale='-'*, *exp='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate a multi-look intensity (MLI) image from an SLC image
Copyright 2018, Gamma Remote Sensing, v4.4 12-Jan-2018 clw/uw/cm

Parameters

- **SLC** – (input) single-look complex image
- **SLC_par** – (input) SLC ISP image parameter file
- **MLI** – (output) multi-look intensity image
- **MLI_par** – (output) MLI ISP image parameter file
- **rlks** – number of range looks
- **azlks** – number of azimuth looks
- **loff** – offset to starting line (enter - for default: 0)

- **nlines** – number of SLC lines to process (enter - for default: entire file)
- **scale** – scale factor for output MLI (enter - for default: 1.0)
- **exp** – exponent for the output MLI (enter - for default: 1.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**multi_look2**(*SLC*, *SLC_par*, *MLI*, *MLI_par*, *r_dec*, *az_dec*, *r_enl='-'*, *az_enl='-'*, *lanczos='-'*, *scale='-'*, *exp='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate an MLI image from an SLC with separate averaging window dimensions and decimation factors
Copyright 2017, Gamma Remote Sensing, v1.0 3-Jan-2018 clw/cm

Parameters

- **SLC** – (input) single-look complex image
- **SLC_par** – (input) SLC ISP image parameter file
- **MLI** – (output) multi-look intensity image
- **MLI_par** – (output) MLI ISP image parameter file
- **r_dec** – range decimation factor (integer)
- **az_dec** – azimuth decimation factor (integer)
- **r_enl** – number of SLC range samples to average, (integer) (enter - for default: r_dec)
- **az_enl** – number of SLC azimuth lines to average, (integer) (enter - for default: az_dec)
- **lanczos** – Lanczos interpolator order 5 -> 9 (enter - for default: 7)
- **scale** – scale factor for output MLI (enter - for default: 1.0)
- **exp** – exponent for the output MLI (enter - for default: 1.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**multi_look_MLI**(*MLI_in*, *MLI_in_par*, *MLI_out*, *MLI_out_par*, *rlks*, *azlks*, *loff='-'*, *nlines='-'*, *scale='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Multi-looking of intensity (MLI) images
Copyright 2017, Gamma Remote Sensing, v1.7 31-Aug-2017 clw/uw/cm

Parameters

- **MLI_in** – (input) multi-look intensity image (MLI) file (float)
- **MLI_in_par** – (input) MLI parameter file
- **MLI_out** – (output) multi-looked MLI image (float)
- **MLI_out_par** – (output) MLI parameter file for output MLI
- **rlks** – range looks for multi-looking
- **azlks** – azimuth looks for multi-looking

- **loff** – offset to starting line (enter - for default = 0)
- **nlines** – number of input MLI lines to process (enter - for default = entire file)
- **scale** – scale factor for output MLI (enter - for default = 1.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**multi_real**(*data_in*, *OFF_par_in*, *data_out*, *OFF_par_out*, *rlks='-'*, *azlks='-'*, *loff='-'*, *nlines='-'*, *roff='-'*, *nsamp='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate multi-look averaged or interpolated 2D image (float data)
Copyright 2012, Gamma Remote Sensing, v2.5 16-Jul-2013 clw/uw

Parameters

- **data_in** – (input) input float image file
- **OFF_par_in** – (input) interferogram/offset parameter file for input image
- **data_out** – (output) output multi-look or interpolated float data file
- **OFF_par_out** – (input/output) interferogram/offset parameter file for output, if already existing, used as input
- **rlks** – number of range looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **azlks** – number azimuth looks, values < -1, interpreted as an image oversampling factor (default: 1)
- **loff** – line offset to starting line (default:0)
- **nlines** – number of lines (default:0, to end of file)
- **roff** – offset to starting range sample (default:0)
- **nsamp** – number of range samples to extract (default:0, to end of line)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**neutron**(*intensity*, *flag*, *width*, *n_thres*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate phase unwrapping neutrons using image intensity
Copyright 2014, Gamma Remote Sensing, v2.3 20-Jan-2014 clw/uw

Parameters

- **intensity** – (input) image intensity
- **flag** – (input) phase unwrapping flag file
- **width** – number of samples/row
- **n_thres** – neutron threshold, multiples of the average intensity (default=6.0)
- **ymin** – offset to starting azimuth row (default = 0)

- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_SLC**(*SLC_1*, *SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *offs*, *snr*, *rwin='-'*, *azwin='-'*, *offsets='-'*, *n_ovr='-'*, *nr='-'*, *naz='-'*, *thres='-'*, *ISZ='-'*, *pflag='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Offsets between SLC images using fringe visibility
Copyright 2016, Gamma Remote Sensing, v2.9 clw 4-Mar-2016

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates (fcomplex)
- **snr** – (output) offset estimation SNR (float)
- **rwin** – search window size (range pixels, (enter - for default from offset parameter file))
- **azwin** – search window size (azimuth lines, (enter - for default from offset parameter file))
- **offsets** – (output) range and azimuth offsets and SNR data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (integer 2**N (1,2,4) default = 2)
- **nr** – number of offset estimates in range direction (enter - for default from offset parameter file)
- **naz** – number of offset estimates in azimuth direction (enter - for default from offset parameter file)
- **thres** – offset estimation quality threshold (enter - for default from offset parameter file)
- **ISZ** – search chip interferogram size (in non-oversampled pixels, default=16)
- **pflag** – print flag (0:print offset summary default=1:print all offset data)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_SLC_tracking**(*SLC_1*, *SLC_2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *offs*, *snr*, *rsw='-'*, *azsw='-'*, *offsets='-'*, *n_ovr='-'*, *thres='-'*, *rstep='-'*, *azstep='-'*, *rstart='-'*, *rstop='-'*, *azstart='-'*, *azstop='-'*, *ISZ='-'*, *pflag='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Offset tracking between SLC images using fringe visibility

Copyright 2016, Gamma Remote Sensing, v3.6 clw 4-Mar-2016

Parameters

- **SLC_1** – (input) single-look complex image 1 (reference)
- **SLC_2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates (fcomplex)
- **snr** – (output) offset estimation SNR (float)
- **rsw** – range search window size (range pixels) (enter - for default from offset parameter file)
- **azsw** – azimuth search window size (azimuth lines) (enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and SNR data in text format, enter - for no output
- **n_ovr** – SLC over-sampling factor (integer 2**N (1,2,4) default: 2)
- **thres** – offset estimation quality threshold (enter - for default from offset parameter file)
- **rstep** – step in range pixels (enter - for default: rsw/2)
- **azstep** – step in azimuth pixels (enter - for default: azsw/2)
- **rstart** – starting range pixel (enter - for default: rsw/2)
- **rstop** – ending range pixel (enter - for default: nr - rsw/2)
- **azstart** – starting azimuth line (enter - for default: azsw/2)
- **azstop** – ending azimuth line (enter - for default: nlines - azsw/2)
- **ISZ** – search chip interferogram size (in non-oversampled pixels, default: 16)
- **pflag** –

  print flag:

  - 0: print offset summary (default)
  - 1: print all offset data
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_add**(*OFF_par1*, *OFF_par2*, *OFF_par3*, *logpath=None*, *outdir=None*, *shellscript=None*)

Add range and azimuth offset polynomial coefficients
Copyright 2008, Gamma Remote Sensing, v1.1 12-Feb-2008 clw

Parameters

- **OFF_par1** – (input) ISP offset/interferogram parameter file
- **OFF_par2** – (input) ISP offset/interferogram parameter file
- **OFF_par3** –

(output) ISP offset/interferogram parameter file with sums of the range and azimuth offset polynomials in OFF_par1 and OFF_par2

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_fit**(*offs*, *ccp*, *OFF_par*, *coffs='-'*, *coffsets='-'*, *thres='-'*, *npoly='-'*, *interact_mode='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Range and azimuth offset polynomial estimation
Copyright 2011, Gamma Remote Sensing, v3.3 28-Nov-2015 clw/uw

Parameters

- **offs** – (input) range and azimuth offset estimates for each patch (fcomplex)
- **ccp** – (input) cross-correlation or SNR of each patch (float)
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **coffs** – (output) culled range and azimuth offset estimates (fcomplex, enter - for none)
- **coffsets** – (output) culled offset estimates and cross-correlation values (text format, enter - for none)
- **thres** – cross-correlation threshold (enter - for default from OFF_par)
- **npoly** – number of model polynomial parameters (enter - for default, 1, 3, 4, 6, default: 4)
- **interact_mode** –

  interactive culling of input data:

  - 0: off (default)
  - 1: on
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_plot_az**(*offset*, *r_min*, *r_max*, *r_plot*, *az_plot*, *logpath=None*, *outdir=None*, *shellscript=None*)

IPTA script: /usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/offset_plot_az
Copyright 2004, Gamma Remote Sensing, v1.3 17-Jan-2005 clw
extract range and azimuth offsets for a range window from an text offset file

Parameters

- **offset** – (input) list of range and azimuth offsets generated by offset_pwr (text)
- **r_min** – minimum range pixel number to extract range and azimuth offsets
- **r_max** – minimum range pixel number to extract range and azimuth offsets
- **r_plot** – range offsets xmgrace plot file
- **az_plot** – azimuth offsets xmgrace plot file

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_plot_r**(*offset*, *az_min*, *az_max*, *r_plot*, *az_plot*, *logpath=None*, *outdir=None*, *shellscript=None*)

IPTA script: /usr/local/GAMMA_SOFTWARE-20180703/ISP/scripts/offset_plot_r
Copyright 2004, Gamma Remote Sensing, v1.3 17-Jan-2005 clw
extract range and azimuth offsets for an azimuth window from an text offset file

Parameters

- **offset** – (input) list of range and azimuth offsets generated by offset_pwr (text)
- **az_min** – minimum azimuth line number to extract range and azimuth offsets
- **az_max** – minimum azimuth line number to extract range and azimuth offsets
- **r_plot** – range offsets xmgrace plot file
- **az_plot** – azimuth offsets xmgrace plot file
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_pwr**(*SLC1*, *SLC2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *offs*, *ccp*, *rwin='-'*, *azwin='-'*, *offsets='-'*, *n_ovr='-'*, *nr='-'*, *naz='-'*, *thres='-'*, *lanczos='-'*, *bw_frac='-'*, *deramp='-'*, *int_filt='-'*, *pflag='-'*, *pltflg='-'*, *ccs='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Offset estimation between SLC images using intensity cross-correlation
Copyright 2017, Gamma Remote Sensing, v5.5 clw/cm 17-Nov-2017

Parameters

- **SLC1** – (input) single-look complex image 1 (reference)
- **SLC2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates in range and azimuth (fcomplex)
- **ccp** – (output) cross-correlation of each patch (0.0->1.0) (float)
- **rwin** – range patch size (range pixels, enter - for default from offset parameter file)
- **azwin** – azimuth patch size (azimuth lines, enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and cross-correlation data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
- **nr** – number of offset estimates in range direction (enter - for default from offset parameter file)

- **naz** – number of offset estimates in azimuth direction (enter - for default from offset parameter file)
- **thres** – cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
- **lanczos** – Lanczos interpolator order 5 -> 9 (enter - for default: 5)
- **bw_frac** – bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
- **deramp** –

  deramp SLC phase flag (enter - for default)

  - 0: no deramp (Doppler centroid close to 0) (default)
  - 1: deramp SLC phase
- **int_filt** –

  intensity low-pass filter flag (enter - for default)

  - 0: no filter
  - 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)
- **pflag** –

  print flag (enter - for default)

  - 0: print offset summary (default)
  - 1: print all offset data
- **pltflg** –

  plotting flag (enter - for default)

  - 0: none (default)
  - 1: screen output
  - 2: screen output and PNG format plots
  - 3: output plots in PDF format
- **ccs** –

  (output) cross-correlation standard deviation of each patch (float)

  - NOTE: ScanSAR and TOPS data need to be previously deramped
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_pwr_tracking**(*SLC1*, *SLC2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *offs*, *ccp*, *rwin='-'*, *azwin='-'*, *offsets='-'*, *n_ovr='-'*, *thres='-'*, *rstep='-'*, *azstep='-'*, *rstart='-'*, *rstop='-'*, *azstart='-'*, *azstop='-'*, *lanczos='-'*, *bw_frac='-'*, *deramp='-'*, *int_filt='-'*, *pflag='-'*, *pltflg='-'*, *ccs='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Offset tracking between SLC images using intensity cross-correlation
Copyright 2017, Gamma Remote Sensing, v5.7 clw/cm 17-Nov-2017

Parameters

- **SLC1** – (input) single-look complex image 1 (reference)
- **SLC2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates in range and azimuth (fcomplex)
- **ccp** – (output) cross-correlation of each patch (0.0->1.0) (float)
- **rwin** – range patch size (range pixels, enter - for default from offset parameter file)
- **azwin** – azimuth patch size (azimuth lines, enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and cross-correlation data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
- **thres** – cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
- **rstep** – step in range pixels (enter - for default: rwin/2)
- **azstep** – step in azimuth pixels (enter - for default: azwin/2)
- **rstart** – offset to starting range pixel (enter - for default: 0)
- **rstop** – offset to ending range pixel (enter - for default: nr-1)
- **azstart** – offset to starting azimuth line (enter - for default: 0)
- **azstop** – offset to ending azimuth line (enter - for default: nlines-1)
- **lanczos** – Lanczos interpolator order 5 -> 9 (enter - for default: 5)
- **bw_frac** – bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
- **deramp** –

  deramp SLC phase flag (enter - for default)

  - 0: no deramp (Doppler centroid close to 0) (default)
  - 1: deramp SLC phase
- **int_filt** –

  intensity low-pass filter flag (enter - for default)

  - 0: no filter
  - 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)
- **pflag** –

  print flag (enter - for default)

  - 0: print offset summary (default)
  - 1: print all offset data
- **pltflg** –

  plotting flag (enter - for default)

  - 0: none (default)
  - 1: screen output

- 2: screen output and PNG format plots
- 3: output plots in PDF format

- **ccs** –

  (output) cross-correlation standard deviation of each patch (float)

  - NOTE: ScanSAR and TOPS data need to be previously deramped

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_pwr_tracking2**(*SLC1*, *SLC2*, *SLC1_par*, *SLC2_par*, *OFF_par*, *offs*, *ccp*, *OFF_par2='-'*, *offs2='-'*, *rwin='-'*, *azwin='-'*, *offsets='-'*, *n_ovr='-'*, *thres='-'*, *rstep='-'*, *azstep='-'*, *rstart='-'*, *rstop='-'*, *azstart='-'*, *azstop='-'*, *bw_frac='-'*, *deramp='-'*, *int_filt='-'*, *pflag='-'*, *pltflg='-'*, *ccs='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Intensity cross-correlation offset tracking with the initial offset for each patch determined from input offset map
Copyright 2017, Gamma Remote Sensing, v1.5 clw/cm 20-Mar-2017

Parameters

- **SLC1** – (input) single-look complex image 1 (reference)
- **SLC2** – (input) single-look complex image 2
- **SLC1_par** – (input) SLC-1 ISP image parameter file
- **SLC2_par** – (input) SLC-2 ISP image parameter file
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs** – (output) offset estimates in range and azimuth (fcomplex)
- **ccp** – (output) cross-correlation of each patch (0.0->1.0) (float)
- **OFF_par2** – (input) ISP offset/interferogram parameter file of the offset map to determine initial offsets (enter - for none)
- **offs2** – (input) input range and azimuth offset map to determine initial offsets (enter - for none)
- **rwin** – range patch size (range pixels, enter - for default from offset parameter file)
- **azwin** – azimuth patch size (azimuth lines, enter - for default from offset parameter file)
- **offsets** – (output) range and azimuth offsets and cross-correlation data in text format, enter - for no output
- **n_ovr** – SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
- **thres** – cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
- **rstep** – step in range pixels (enter - for default: rwin/2)
- **azstep** – step in azimuth pixels (enter - for default: azwin/2)
- **rstart** – offset to starting range pixel (enter - for default: 0)
- **rstop** – offset to ending range pixel (enter - for default: nr-1)

- **azstart** – offset to starting azimuth line (enter - for default: 0)
- **azstop** – offset to ending azimuth line (enter - for default: nlines-1)
- **bw_frac** – bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
- **deramp** –

  deramp SLC phase flag (enter - for default)

  - 0: no deramp (Doppler centroid close to 0) (default)
  - 1: deramp SLC phase
- **int_filt** –

  intensity low-pass filter flag (enter - for default)

  - 0: no filter
  - 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)
- **pflag** –

  print flag (enter - for default)

  - 0: print offset summary (default)
  - 1: print all offset data
- **pltflg** –

  plotting flag (enter - for default)

  - 0: none (default)
  - 1: screen output
  - 2: screen output and PNG format plots
  - 3: output plots in PDF format
- **ccs** –

  (output) cross-correlation standard deviation of each patch (float)

  - NOTE: ScanSAR and TOPS data need to be previously deramped
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_sub**(*offs*, *OFF_par*, *offs_sub*, *logpath=None*, *outdir=None*, *shellscript=None*)

Subtraction of polynomial from range and azimuth offset estimates
Copyright 2017, Gamma Remote Sensing, v1.0 27-Mar-2017 cm

Parameters

- **offs** – (input) range and azimuth offset estimates (fcomplex)
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **offs_sub** – (output) range and azimuth offset estimates after polynomial subtraction (fcomplex)
- **logpath** (*str or None*) – a directory to write command logfiles to

- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**offset_tracking**(*offs*, *ccp*, *SLC_par*, *OFF_par*, *disp_map*, *disp_val='-'*, *mode='-'*, *thres='-'*, *poly_flag='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Conversion of range and azimuth offsets files to displacement map
Copyright 2017, Gamma Remote Sensing, v2.0 4-Apr-2017 ts/clw/uw

Parameters

- **offs** – (input) range and azimuth offset estimates (fcomplex)
- **ccp** – (input) cross-correlation of the offset estimates (float)
- **SLC_par** – (input) SLC parameter file of reference SLC
- **OFF_par** – (input) offset parameter file used in the offset tracking
- **disp_map** – (output) range and azimuth displacement estimates (fcomplex)
- **disp_val** – (output) range and azimuth displacement estimates and cross-correlation values (enter - for none) (text)
- **mode** –

  flag indicating displacement mode:

  - 0: displacement in range and azimuth pixels
  - 1: displacement in meters in slant range and azimuth directions
  - 2: displacement in meters in ground range and azimuth directions (default)
- **thres** – cross-correlation threshold to accept offset value (default from OFF_par)
- **poly_flag** –

  flag indicating if trend calculated using offset polynomials from OFF_par is subtracted:

  - 0: do not subtract polynomial trend from offset data
  - 1: subtract polynomial trend from offset data (default)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ACS_ERS**(*CEOS_SAR_leader*, *SLC_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file generation for ERS SLC data from the ACS processor
Copyright 2005, Gamma Remote Sensing, v1.3 17-Oct-2005 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in

- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ASAR**(*ASAR_ERS_file*, *output_name*, *K_dB='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Extract SLC/MLI image parameters and images from ENVISAT ASAR SLC, WSS, APP, and PRI products
Copyright 2018, Gamma Remote Sensing, v2.7 9-Apr-2018 clw/uw/awi

Parameters

- **ASAR_ERS_file** – (input) ASAR or ERS data in ASAR format (SAR_IMS_1P) including header and image as provided by ESA
- **output_name** – (output) common part of output file names (e.g. YYYMMDD date)
- **K_dB** –

  Calibration factor in dB (nominal value for all ASAR modes: 55.0)

  - NOTE: Use - to use the calibration factor provided in the ASAR file header
  - NOTE: In the case that a calibration factor is specified on the command line, PRI images are converted to radiometrically calibrated ground-range intensity images in float format
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ASF_91**(*CEOS_leader*, *CEOS_trailer*, *SLC_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

SLC parameter file for data data from theAlaska SAR Facility (1991-1996)
Copyright 2008, Gamma Remote Sensing, v3.3 25-Mar-2008 clw/uw

Parameters

- **CEOS_leader** – (input) ASF CEOS leader file
- **CEOS_trailer** – (input) ASF CEOS trailer file
- **SLC_par** – (output) ISP SLC image parameter file
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ASF_96**(*CEOS_SAR_leader*, *SLC_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for ASF data 1996–>present v1.1
Copyright 2003, Gamma Remote Sensing, v1.4 4-Aug-2003 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to

- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ASF_PRI**(*CEOS_leader*, *CEOS_data*, *GRD_par*, *GRD*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for ASF detected ground range images (L1) Sep 1996 –> present
Copyright 2014, Gamma Remote Sensing, v1.3 3-Apr-2014 clw/uw

Parameters

- **CEOS_leader** – (input) CEOS leader file
- **CEOS_data** – (input) CEOS data file binary)
- **GRD_par** – (output) ISP ground range image parameter file
- **GRD** –

  (output) ISP ground range image (enter - for none, float intensity)

  - NOTE: The input data converted to intensity using the expression: (dn/1000.)**2
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ASF_RSAT_SS**(*CEOS_leader*, *CEOS_data*, *GRD_par*, *GRD*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for ASF Radarsat-1 SCANSAR images
Copyright 2004, Gamma Remote Sensing, v1.0 27-Aug-2004 clw/uw

Parameters

- **CEOS_leader** – (input) CEOS leader file (Radarsat-1 SCANSAR)
- **CEOS_data** – (input) CEOS data file (Radarsat-1 SCANSAR)
- **GRD_par** – (output) ISP image parameter file (example <orbit>.mli.par)
- **GRD** – (output) ISP image (example <orbit>.mli) (enter - for none, short integer)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ASF_SLC**(*CEOS_leader*, *SLC_par*, *CEOS_data='-'*, *SLC='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC image parameter file and reformat data
Copyright 2012, Gamma Remote Sensing, v1.0 27-Aug-2012 clw/uw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <date>.slc.par)
- **CEOS_data** – (input) CEOS data file (example: dat_01.001)

- **SLC** – (output) SLC data with file and line headers removed (example: <date>.slc)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ATLSCI_ERS**(*CEOS_SAR_leader*, *CEOS_Image*, *SLC_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for ATL-SCI ERS SLC data
Copyright 2003, Gamma Remote Sensing, v2.8 24-Nov-2003 clw

Parameters

- **CEOS_SAR_leader** – (input) CEOS SAR leader file (LEA_01.001)
- **CEOS_Image** – (input) CEOS image data segment (DAT_01.001)
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_CS_SLC**(*HDF5*, *trunk*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate ISP SLC parameter and image files for Cosmo-Skymed SCS data
Copyright 2017, Gamma Remote Sensing, v1.8 26-Sep-2017 awi/ms/cw/uw

Parameters

- **HDF5** – (input) SCS data file in HDF5 format
- **trunk** –

  (output) output file name trunk used for output filenames
  (example: yyyymmdd -> yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_CS_SLC_TIF**(*GeoTIFF*, *XML*, *trunk*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate ISP SLC parameter and image files for Cosmo Skymed SCS data in GeoTIFF format
Copyright 2018, Gamma Remote Sensing, v1.5 7-Feb-2018 awi/ms/clw/cm

Parameters

- **GeoTIFF** – (input) SCS data file in GeoTIFF format
- **XML** – (input) SCS meta data file in XML format
- **trunk** –

(output) output file name trunk used for output filenames
(example: yyyymmdd -> yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.slc.par)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_EORC_JERS_SLC**(*CEOS_SAR_leader*, *SLC_par*, *CEOS_data='-'*, *SLC='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Reformat EORC processed JERS-1 SLC and generate the ISP parameter file
Copyright 2008, Gamma Remote Sensing, v1.4 9-Oct-2008 clw

Parameters

- **CEOS_SAR_leader** – (input) CEOS SAR leader file for JERS SLC processed by EORC
- **SLC_par** – (output) ISP image parameter file
- **CEOS_data** – (input) CEOS format SLC data (IMOP_01.DAT, enter - for none)
- **SLC** – (output) reformated JERS SLC (example: yyyymmdd.SLC, enter - for none)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_EORC_PALSAR**(*CEOS_leader*, *SLC_par*, *CEOS_data*, *SLC='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Reformat EORC PALSAR + PALSAR2 level 1.1 CEOS format SLC data and generate the ISP parameter file
Copyright 2017, Gamma Remote Sensing, v2.8 18-Dec-2017 clw

Parameters

- **CEOS_leader** – (input) CEOS leader file for PALSAR or PALSAR-2 Level 1.1 SLC data (LED...)
- **SLC_par** – (output) ISP image parameter file (example: yyyymmdd.slc.par)
- **CEOS_data** – (input) PALSAR CEOS format Level 1.1 SLC (IMG...)
- **SLC** – (output) reformatted PALSAR SLC (example: yyyymmdd.slc, enter - for none)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ERSDAC_PALSAR**(*ERSDAC_SLC_par*, *SLC_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate the ISP image parameter file from ERSDAC PALSAR level 1.1 SLC data
Copyright 2011, Gamma Remote Sensing, v1.5 31-Oct-2011 clw

Parameters

- **ERSDAC_SLC_par** – (input) ERSDAC SLC parameter file Level 1.1 (PASL11*.SLC.par)
- **SLC_par** – (output) ISP image parameter file (example: yyyymmdd.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_ESA_ERS**(*CEOS_SAR_leader*, *SLC_par*, *inlist*, *CEOS_DAT='-'*, *SLC='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file generation for ERS SLC data from the PGS, VMP, and SPF processors
Copyright 2012, Gamma Remote Sensing, v1.4 12-Jan-2012 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example: <date>.slc.par)
- **inlist** – a list of arguments to be passed to stdin
- **CEOS_DAT** – (input) CEOS data file (example: DAT_01.001)
- **SLC** – (output) SLC data with file and line headers removed (example: <date>.slc)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_GF3_SLC**(*GeoTIFF*, *annotation_XML*, *SLC_par*, *SLC='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC parameter file and SLC image from a Gaofen-3 data set in GeoTIFF format
Copyright 2018, Gamma Remote Sensing, v1.2 8-Mar-2018 cm

Parameters

- **GeoTIFF** – (input) Gaofen-3 data file in GeoTIFF format (*.tiff) (enter - for none)
- **annotation_XML** – (input) Gaofen-3 annotation file in XML format (*.meta.xml)
- **SLC_par** – (output) ISP SLC parameter file (example: yyyymmdd.slc.par)
- **SLC** – (output) ISP SLC data file (example: yyyymmdd.slc) (enter - for none, SLC output will not be produced)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_IECAS_SLC**(*aux_data*, *slc_Re*, *slc_Im*, *date*, *SLC_par*, *SLC*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC parameter and image files for IECAS SLC data
Copyright 2011, Gamma Remote Sensing, v1.2 24-Jan-2011

Parameters

- **aux_data** – (input) IECAS SAR auxillary data (POS*.dat)
- **slc_Re** – (input) real part of complex SLC data
- **slc_Im** – (input) imaginary part of complex SLC data
- **date** – (input) acquistion date format: YYYYMMDD (example 20110121) from aux_data filename
- **SLC_par** – (output) ISP SLC parameter file
- **SLC** – (output) SLC image
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_KC_PALSAR_slr**(*facter_m*, *CEOS_leader*, *SLC_par*, *pol*, *pls_mode*, *KC_data*, *pwr*, *fdtab='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate ISP parameter file, Doppler table, and images for PALSAR KC Slant-Range data
Copyright 2013, Gamma Remote Sensing, v1.9.1 20-Aug-2013 ms,awi,clw

Parameters

- **facter_m** – (input) PALSAR Kyoto-Carbon parameter file
- **CEOS_leader** – (input) PALSAR Kyoto-Carbon leader file (LED)
- **SLC_par** – (output) ISP image parameter file (example: yyyymmdd.mli.par)
- **pol** – polarization e.g. HH or HV
- **pls_mode** –

  PALSAR acquisition mode:

  - 1: Fine Beam Single
  - 2: Fine Beam Double
  - 3: Wide Beam
- **KC_data** – (input) PALSAR Kyoto-Carbon data (short, little endian, amplitude)
- **pwr** – (output) PALSAR intensity (float, GAMMA Software endianness)
- **fdtab** – (output)table of output polynomials, one polynomial/block used as input to gc_map_fd
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_KS_DGM**(*HDF5*, *trunk*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate ISP SLC parameter and PRI image files for Kompsat DGM data
Copyright 2018, Gamma Remote Sensing, v1.1 7-Feb-2018 awi/cm

Parameters

- **HDF5** – (input) DGM data file in HDF5 format
- **trunk** –

(output) output file name trunk used for output filenames
(example: yyyymmdd -> yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.pri.par)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_KS_SLC**(*HDF5*, *trunk*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate ISP SLC parameter and image files for Kompsat SCS data
Copyright 2018, Gamma Remote Sensing, v1.5 7-Feb-2018 awi/clw/cm

Parameters

- **HDF5** – (input) SCS data file in HDF5 format
- **trunk** –

  (output) output file name trunk used for output filenames
  (example: yyyymmdd -> yyyymmdd_pol_beamid.slc yyyymmdd_pol_beamid.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_MSP**(*SAR_par*, *PROC_par*, *SLC_MLI_par*, *image_format='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP image parameter file from MSP processing parameter and sensor files
Copyright 2010, Gamma Remote Sensing, v3.3 2-Oct-2010 clw/uw

Parameters

- **SAR_par** – (input) MSP SAR sensor parameter file
- **PROC_par** – (input) MSP processing parameter file
- **SLC_MLI_par** – (output) ISP SLC/MLI image parameter file
- **image_format** –

  image format flag (default: from MSP processing parameter file)

  - 0: fcomplex (pairs of 4-byte float)
  - 1: scomplex (pairs of 2-byte short integer)
  - 2: float (4-bytes/value)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_PRI**(*CEOS_SAR_leader*, *PRI_par*, *CEOS_DAT*, *PRI*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file generation for ERS PRI data from the PGS and VMP processors

Copyright 2012, Gamma Remote Sensing, v1.6 12-Jan-2012 clw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file for PRI product
- **PRI_par** – (output) ISP image parameter file (example: <yyyymmdd>.pri.par)
- **CEOS_DAT** – (input) CEOS data file (example: DAT_01.001)
- **PRI** – (output) PRI data with file and line headers removed (example: <yyyymmdd>.pri)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_PRI_ESRIN_JERS**(*CEOS_SAR_leader*, *PRI_par*, *CEOS_DAT*, *PRI*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP GRD parameter file for ESRIN processed JERS PRI data
Copyright 2008, Gamma Remote Sensing, v1.8 16-May-2008 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file for PRI product
- **PRI_par** – (output) ISP image parameter file (example: <yyyymmdd>.pri.par)
- **CEOS_DAT** – (input) CEOS data file (example: DAT_01.001)
- **PRI** – (output) PRI data with file and line headers removed (example: <yyyymmdd>.pri)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_PulSAR**(*CEOS_SAR_leader*, *SLC_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file generation for ERS SLC data from the PULSAR SAR processor
Copyright 2003, Gamma Remote Sensing, v1.2 4-Aug-2003 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RISAT_GRD**(*CEOS_leader*, *BAND_META*, *GRD_par*, *CEOS_image*, *GRD='-'*, *line_dir='-'*, *pix_dir='-'*, *cal_flg='-'*, *KdB='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Read RISAT-1 Ground-Range data from a CEOS data set and perform radiometric calibration
Copyright 2015, Gamma Remote Sensing, v1.2 24-Feb-2015 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file (example: lea_01.001)
- **BAND_META** – (input) BAND_META.txt, additional RISAT system parameters for the scene (format keywork=value)
- **GRD_par** – (output) ISP GRD parameter file (example: YYYYMMDD.grd.par)
- **CEOS_image** – (input) CEOS Ground-Range image file (example: dat_01.001)
- **GRD** – (output) Ground-Range data with file and line headers removed (enter - for none: example: YYYYMMDD.grd)
- **line_dir** –

  set output image line direction (enter - for default):

  - 0: used value derived from CEOS leader file
  - 1: retain input data line direction (default)
  - -1: reverse input data line direction
- **pix_dir** –

  set output pixel direction (enter - for default):

  - 0: used value derived from CEOS leader file
  - 1: retain input data pixel direction (default)
  - -1: reverse input data pixel direction
- **cal_flg** –

  calibration flag (enter - for default):

  - 0: do not apply radiometric calibration
  - 1: apply radiometric calibration including KdB and incidence angle correction (default)
- **KdB** – calibration constant (dB) (enter - to use value in the CEOS leader)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RISAT_SLC**(*CEOS_leader*, *BAND_META*, *SLC_par*, *CEOS_image*, *SLC='-'*, *line_dir='-'*, *pix_dir='-'*, *cal_flg='-'*, *KdB='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Read RISAT-1 CEOS format SLC data and perform radiometric calibration
Copyright 2013, Gamma Remote Sensing, v1.1 3-Jun-2013 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file (example: lea_01.001)
- **BAND_META** – (input) BAND_META.txt, additional RISAT system parameters for the scene (format keywork=value)
- **SLC_par** – (output) ISP SLC image parameter file (example: YYYYMMDD.grd.par)
- **CEOS_image** – (input) CEOS SLC image file (example: dat_01.001)
- **SLC** – (output) SLC data with file and line headers removed (enter - for none: example: YYYYMMDD.grd)

- **line_dir** –

  set output image line direction (enter - for default):

  - 0: used value derived from CEOS leader file
  - 1: retain input data line direction (default)
  - -1: reverse input data line direction

- **pix_dir** –

  set output pixel direction (enter - for default):

  - 0: used value derived from CEOS leader file
  - 1: retain input data pixel direction (default)
  - -1: reverse input data pixel direction

- **cal_flg** –

  calibration flag (enter - for default):

  - 0: do not apply radiometric calibration
  - 1: apply radiometric calibration including KdB and incidence angle correction (default)

- **KdB** – calibration constant (dB) (enter - to use value in the CEOS leader)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RSAT2_SG**(*product_XML*, *lut_XML*, *GeoTIFF*, *polarization*, *GRD_par*, *GRD*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC parameter and ground range image files for Radarsat 2 SGF/SGX data
Copyright 2018, Gamma Remote Sensing, v1.9 7-Feb-2018 awi/cw/cm

Parameters

- **product_XML** – (input) Radarsat-2 product annotation XML file (product.xml)
- **lut_XML** – (input) Radarsat-2 calibration XML file (lutSigma.xml), use - for no calibration
- **GeoTIFF** – (input) image data file in GeoTIFF format (imagery_PP.tif)
- **polarization** – (input) image polarization: HH, VV, HV, VH
- **GRD_par** – (output) ISP GRD parameter file (example: yyyymmdd_PP.grd.par)
- **GRD** – (output) float GRD data file (example: yyyymmdd_pp.grd)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RSAT2_SLC**(*product_XML*, *lut_XML*, *GeoTIFF*, *polarization*, *SLC_par*, *SLC*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC parameter and image files for Radarsat 2 SLC data from GeoTIFF
Copyright 2018, Gamma Remote Sensing, v2.6 7-Feb-2018 awi/clw/cm

Parameters

- **product_XML** – (input) Radarsat-2 product annotation XML file (product.xml)
- **lut_XML** – (input) Radarsat-2 calibration XML file (lutSigma.xml), use - for no calibration
- **GeoTIFF** – (input) image data file in GeoTIFF format (imagery_PP.tif)
- **polarization** – (input) image polarization: HH, VV, HV, VH
- **SLC_par** – (output) ISP SLC parameter file (example: yyyymmdd_pp.slc.par)
- **SLC** – (output) SLC data file (example: yyyymmdd_pp.slc)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RSAT_SCW**(*CEOS_leader*, *CEOS_trailer*, *CEOS_data*, *GRD_par*, *GRD*, *sc_dB='-'*, *dt='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for SCANSAR Wide Swath Data
Copyright 2012, Gamma Remote Sensing, v2.0 14-Feb-2012 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file
- **CEOS_trailer** – (input) CEOS SAR trailer file
- **CEOS_data** – (input) CEOS data file binary)
- **GRD_par** – (output) ISP ground range image parameter file (example <orbit>.mli.par)
- **GRD** – (output) ISP ground range image (example <orbit>.mli) (enter - for none, float)
- **sc_dB** – intensity scale factor in dB (enter - for default: 0.00)
- **dt** – azimuth image time offset (s) (enter - for default = 0.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RSAT_SGF**(*CEOS_leader*, *CEOS_data*, *GRD_par*, *GRD*, *sc_dB='-'*, *dt='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for RSI/Atlantis Radarsat SGF (ground range) and SCANSAR SCW16 data
Copyright 2012, Gamma Remote Sensing, v2.2 14-Feb-2012 clw

Parameters

- **CEOS_leader** – (input) CEOS leader file (RSI SGF or SCW16 products, LEA_01.001)
- **CEOS_data** – (input) CEOS data file (RSI SGF or SCW16 products, DAT_01.001)
- **GRD_par** – (output) ISP ground range image parameter file (example <orbit>.mli.par)
- **GRD** – (output) ISP ground range image (example <orbit>.grd.par) (enter - for none, float)
- **sc_dB** – intensity scale factor in dB (enter - for default: 0.00)
- **dt** – azimuth image time offset (s) (enter - for default = 0.0)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RSAT_SLC**(*CEOS_leader*, *SLC_par*, *CEOS_data*, *SLC='-'*, *sc_dB='-'*, *dt='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for RSI/Atlantis/ASF processed Radarsat SLC data
Copyright 2012, Gamma Remote Sensing, v4.0 5-Sep-2012 clw

Parameters

- **CEOS_leader** – (input) CEOS SAR leader file (example: lea_01.001)
- **SLC_par** – (output) ISP SLC parameter file (example: <date>.slc.par)
- **CEOS_data** – (input) CEOS data file (example: dat_01.001)
- **SLC** – (output) SLC data with file and line headers removed (example: <date>.slc)
- **sc_dB** – intensity scale factor in dB (enter - for default: 60.00)
- **dt** – azimuth image time offset (s) (enter - for default = 0.0)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_RSI_ERS**(*CEOS_SAR_leader*, *SLC_par*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP parameter file for RSI processed ERS SLC data
Copyright 2003, Gamma Remote Sensing, v1.7 4-Aug-2003 clw/uw

Parameters

- **CEOS_SAR_leader** – (input) ERS CEOS SAR leader file
- **SLC_par** – (output) ISP SLC parameter file (example <orbit>.slc.par)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_S1_GRD**(*GeoTIFF*, *annotation_XML*, *calibration_XML*, *noise_XML*, *MLI_par*, *MLI*, *GRD_par='-'*, *GRD='-'*, *eflg='-'*, *rps='-'*, *noise_pwr='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate MLI and GRD images and parameter files from a Sentinel-1 GRD product
Copyright 2018, Gamma Remote Sensing, v3.2 30-Apr-2018 awi/clw/ts/cm

Parameters

- **GeoTIFF** – (input) image data file in GeoTIFF format (enter - for none, *.tiff)
- **annotation_XML** – (input) Sentinel-1 L1 XML annotation file
- **calibration_XML** – (input) Sentinel-1 L1 radiometric calibration XML file (enter - for no radiometric calibration)

- **noise_XML** – (input) Sentinel-1 L1 noise XML file (enter - to not subtract thermal noise power level)
- **MLI_par** – (output) MLI parameter file (example: yyyymmdd_pp.mli.par)
- **MLI** – (output) MLI data file in slant range geometry (example: yyyymmdd_pp.mli, enter - for none)
- **GRD_par** – (output) GRD parameter file (example: yyyymmdd_pp.grd.par, enter - for none)
- **GRD** – (output) GRD data file (example: yyyymmdd_pp.grd, enter - for none)
- **eflg** –

  GR-SR grid extrapolation flag:

  - 0: no extrapolation of the GR-SR grid beyond the grid boundaries
  - 1: permit extrapolation of the GR-SR grid to cover the entire image (default)
  - NOTE: extrapolation of the GR-SR grid may introduce geocoding errors
- **rps** – slant range pixel spacing (m) (enter - for default: calculated from ground-range parameters)
- **noise_pwr** –

  noise intensity for each MLI sample in slant range using data from noise_XML

  - NOTE: when the noise_pwr file is specified, noise power correction will NOT be applied to the MLI data values
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_S1_SLC**(*GeoTIFF*, *annotation_XML*, *calibration_XML*, *noise_XML*, *SLC_par*, *SLC*, *TOPS_par='-'*, *dtype='-'*, *sc_dB='-'*, *noise_pwr='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC parameter and image files for Sentinel-1 SLC data
Copyright 2018, Gamma Remote Sensing, v4.0 30-Apr-2018 awi/clw/cm

Parameters

- **GeoTIFF** – (input) image data file in GeoTIFF format (enter - for none, *.tiff)
- **annotation_XML** – (input) Sentinel-1 L1 XML annotation file
- **calibration_XML** – (input) Sentinel-1 L1 radiometric calibration XML file (enter - for no radiometric calibration)
- **noise_XML** – (input) Sentinel-1 L1 noise XML file (enter - to not subtract thermal noise power level)
- **SLC_par** – (output) ISP SLC parameter file (example: yyyymmdd_iw1_vv.slc.par)
- **SLC** – (output) SLC data file (enter - for none, example: yyyymmdd_iw1_vv.slc)
- **TOPS_par** – (output) SLC burst annotation file, TOPS and EW SLC data only (enter - for none, example: yyyymmdd_iw1_vv.tops_par)
- **dtype** –

  output data type:

  - 0: FCOMPLEX (default)

- 1: SCOMPLEX

- **sc_dB** – scale factor for FCOMPLEX -> SCOMPLEX, (enter - for default: HH,VV (dB): 60.0000, VH,HV: 70.0000)
- **noise_pwr** –

  noise intensity for each SLC sample in slant range using data from noise_XML

  - NOTE: when the noise_pwr file is specified, noise power will NOT be subtracted from the image data values
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_SIRC**(*CEOS_leader*, *SLC_par*, *UTC_MET='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP SLC parameter file from SIR-C CEOS leader file
Copyright 2009, Gamma Remote Sensing, v2.5 30-Oct-2009 clw/uw

Parameters

- **CEOS_leader** – (input) JPL SIR-C CEOS leader file
- **SLC_par** – (output) ISP SLC parameter file
- **UTC_MET** –

  time reference for state vectors:
  MET (Mission Elapsed Time) or UTC (default=UTC)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_TX_GRD**(*annotation_XML*, *GeoTIFF*, *GRD_par*, *GRD*, *pol='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate ground range image and image parameter file for Terrasar-X MGD data in GeoTIFF format
Copyright 2014, Gamma Remote Sensing, v1.3 17-Oct awi/clw

Parameters

- **annotation_XML** – (input) Terrasar-X product annotation XML file
- **GeoTIFF** –

  (input) image data file in geotiff format

  - NOTE: make sure the data set contains the selected polarisation)
- **GRD_par** – ISP ground range image parameter file (example: yyyymmdd.grd.par
- **GRD** – (output) calibrated ground range data file (example: yyyymmdd.grd)
- **pol** – polarisation: HH, HV, VH, VV (default: first polarisation found in the annotation_XML)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in

- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_TX_SLC**(*annotation_XML*, *COSAR*, *SLC_par*, *SLC*, *pol='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC parameter file and SLC image from a Terrasar-X SSC data set
Copyright 2017, Gamma Remote Sensing, v2.3 7-Mar-2017 awi/clw

Parameters

- **annotation_XML** – (input) TerraSAR-X product annotation XML file
- **COSAR** – (input) COSAR SSC stripmap or spotlight mode SLC data file
- **SLC_par** – (output) ISP SLC parameter file (example: yyyymmdd.slc.par)
- **SLC** – (output) SLC data file, example: yyyymmdd.slc (enter - for none, SLC output will not be produced)
- **pol** – polarisation HH, HV, VH, VV (default: first polarisation found in the annotation_XML)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_TX_ScanSAR**(*annot_XML*, *swath*, *SLC_par*, *SLC*, *TOPS_par*, *bwflg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate SLC, SLC_par and TOPS_par from a Terrasar-X ScanSAR data set
Copyright 2018, Gamma Remote Sensing, v1.6 12-Feb-2018 clw/cm/awi

Parameters

- **annot_XML** –

  (input) TerraSAR-X ScanSAR product annotation XML file including path

  - NOTE: The path to the image products is determined from the path to the XML annotation

- **swath** –

  number specifying the desired ScanSAR swath (1 -> maximum number of swaths (4 or 6))

  - NOTE: The image product name is specified in the XML file

- **SLC_par** – (output) ISP SLC parameter file (example: yyyymmdd.slc.par)
- **SLC** – (output) SLC ScanSAR data file, example: yyyymmdd.slc (enter - for none, SLC output will not be produced)
- **TOPS_par** – (output) SLC ScanSAR burst annotation file (example: yyyymmdd_s1.tops_par
- **bwflg** –

  burst window flag:

  - 0: use first and last annotation line values specified in the annot_XML
  - 1: extend first and last valid line to include all data lines (default)

  - NOTE: While TSX ScanSAR data are not acquired in TOPS mode, the same data structure can be used for burst annotation
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**par_UAVSAR_SLC**(*ann*, *SLC_MLI_par*, *image_type*, *image_format*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP image parameter file from UAVSAR annotation file (ann) for SLC and MLC products
Copyright 2014, Gamma Remote Sensing, v1.3 20-Aug-2014 clw

Parameters

- **ann** – (input) UAVSAR annotation file (*ann.txt)
- **SLC_MLI_par** – (output) ISP image parameter file
- **image_type** –

  image type flag

  - 0: SLC (slc) in slant range coordinates
  - 1: MLC (mlc) in slant range coordinates HHHH*, VVVV*, HVHV* are FLOAT format HHHV*, HHVV*, HVVV* are FCOMPLEX format
- **image_format** –

  image data format flag

  - 0: FCOMPLEX (pairs of 4-byte float (re,im))
  - 2: FLOAT (4-bytes/value)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ph_slope_base**(*int_in*, *SLC_par*, *OFF_par*, *base*, *int_out*, *int_type='-'*, *inverse='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Subtract/add interferogram flat-Earth phase trend as estimated from initial baseline
Copyright 2006, Gamma Remote Sensing, v4.4 3-Nov-2006 clw

Parameters

- **int_in** – (input) interferogram (FCOMPLEX) or unwrapped phase (FLOAT) (unflattened)
- **SLC_par** – (input) ISP parameter file for the reference SLC
- **OFF_par** – (input) ISP offset/interferogram parameter file
- **base** – (input) baseline file
- **int_out** – (output) interferogram (FCOMPLEX) or unwrapped phase (FLOAT) with phase trend subtracted/added
- **int_type** – interferogram type: 0=unwrapped phase, 1=complex interf. (default=1)
- **inverse** – subtract/add inversion flag (0=subtract phase ramp, 1=add phase ramp (default=0)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**phase_slope**(*interf*, *slopes*, *width*, *win_sz='-'*, *thres='-'*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate interferogram phase slopes in range and azimuth
Copyright 2011, Gamma Remote Sensing, v1.3 19-Apr-2011 clw/uw

Parameters

- **interf** – (input) interferogram (fcomplex)
- **slopes** – (output) range and azimuth phase slopes (fcomplex)
- **width** – number of samples/row
- **win_sz** – size of region used for slopes determination (default = 5)
- **thres** – correlation threshold for accepting slope estimates 0.0 -> 1.0 (default=.4)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row offset (default = 0)
- **ymax** – last azimuth row offset (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ptarg_SLC**(*SLC_par*, *SLC*, *r_samp*, *az_samp*, *ptr_image*, *r_plot*, *az_plot*, *ptr_par='-'*, *osf='-'*, *win='-'*, *pltflg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Point target response analysis and interpolation for SLC images
Copyright 2016, Gamma Remote Sensing, v1.9 19-Feb-2016 clw

Parameters

- **SLC_par** – (input) SLC image parameter file
- **SLC** – (input) SLC image in FCOMPLEX or SCOMPLEX format
- **r_samp** – point target range sample number
- **az_samp** – point target azimuth line number
- **ptr_image** – (output) oversampled point target image (fcomplex, 1024x1024 samples), with and without phase gradient
- **r_plot** – (output) range point target response plot data (text format)
- **az_plot** – (output) azimuth point target response plot data (text format)
- **ptr_par** – (output) measured point target parameters (text format)
- **osf** – image over-sampling factor, 2, 4, 8, 16, 32, 64 (enter - for default: 16)
- **win** – maximum search window offset (samples) (enter - for default: 1)

- **pltflg** –

  plotting mode flag:

  - 0: none
  - 1: output plots in PNG format (default)
  - 2: screen output
  - 3: output plots in PDF format

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ptarg_cal_MLI**(*MLI_par*, *MLI*, *r_samp*, *az_samp*, *psigma*, *c_r_samp*, *c_az_samp*, *ptr_image*, *r_plot*, *az_plot*, *pcal*, *osf='-'*, *win='-'*, *pltflg='-'*, *psz='-'*, *csz='-'*, *theta_inc='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Point target analysis and radiometric calibration of slant-range and ground-range (GRD) images
Copyright 2016, Gamma Remote Sensing, v2.6 19-Feb-2016 clw

Parameters

- **MLI_par** – (input) slant-range or ground-range image parameter file for detected intensity data
- **MLI** – (input) ground-range or slant range detected image in FLOAT format
- **r_samp** – point target range sample number, target region size is 16x16
- **az_samp** – point target azimuth line number, target region size is 16x16
- **psigma** – radar cross-section of the calibration target in m**2
- **c_r_samp** – clutter region center range sample number, clutter region size is 16x16
- **c_az_samp** – clutter region center azimuth line number, clutter region size is 16x16
- **ptr_image** – (output) oversampled point target image, with and without phase gradient, nominal width: 256
- **r_plot** – (output) range point target response plot data (text format)
- **az_plot** – (output) azimuth point target response plot data (text format)
- **pcal** – (output) measured point target parameters and radiometric calibration factor (text format)
- **osf** – image over-sampling factor, 2, 4, 8, 16, 32, 64 (enter - for default: 16)
- **win** – maximum search window offset (samples) (enter - for default: 1)
- **pltflg** –

  plotting mode flag:

  - 0: none
  - 1: output plots in PNG format (default)
  - 2: screen output
  - 3: output plots in PDF format

- **psz** – point target region size (samples) (enter - for default: 16)
- **csz** – clutter region size (samples) (enter - for default: 16)

- **theta_inc** – incidence angle required for calibration of terrain corrrected RISAT-1 images
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**ptarg_cal_SLC**(*SLC_par*, *SLC*, *r_samp*, *az_samp*, *psigma*, *c_r_samp*, *c_az_samp*, *ptr_image*, *r_plot*, *az_plot*, *pcal*, *osf='-'*, *win='-'*, *pltflg='-'*, *psz='-'*, *csz='-'*, *c_image='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Point target analysis and radiometric calibration of SLC images
Copyright 2016, Gamma Remote Sensing, v2.4 19-Feb-2016 clw

Parameters

- **SLC_par** – (input) SLC image parameter file
- **SLC** – (input) SLC image in FCOMPLEX or SCOMPLEX format
- **r_samp** – point target range sample number, target region size is 16x16
- **az_samp** – point target azimuth line number, target region size is 16x16
- **psigma** – radar cross-section of the calibration target in m**2
- **c_r_samp** – clutter region center range sample number, clutter region size is 16x16
- **c_az_samp** – clutter region center azimuth line number, clutter region size is 16x16
- **ptr_image** – (output) oversampled point target image, with and without phase gradient, nominal width: 256
- **r_plot** – (output) range point target response plot data (text format)
- **az_plot** – (output) azimuth point target response plot data (text format)
- **pcal** – (output) measured point target parameters and radiometric calibration factor (text format)
- **osf** – image over-sampling factor, 2, 4, 8, 16, 32, 64 (enter - for default: 16)
- **win** – maximum search window offset (samples) (enter - for default: 1)
- **pltflg** –

  plotting mode flag:

  - 0: none
  - 1: output plots in PNG format (default)
  - 2: screen output
  - 3: output plots in PDF format
- **psz** – point target region size (samples) (enter - for default: 16)
- **csz** – clutter region size (samples) (enter - for default: 16)
- **c_image** – (output) clutter region image (FCOMPLEX format)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**pwr2ras**(*MLI_tab*, *width*, *scale='-'*, *exp='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate raster images of MLI image files
Copyright 2012, Gamma Remote Sensing, v1.2 27-Nov-2012 clw

Parameters

- **MLI_tab** – (input) list of mli intensity images (float)
- **width** – image width of input data files
- **scale** – image display scale factor (enter - for default, default: 0.9)
- **exp** – image display exponent (default: 0.35)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**radcal_MLI**(*MLI*, *MLI_par*, *OFF_par*, *CMLI*, *antenna='-'*, *rloss_flag='-'*, *ant_flag='-'*, *refarea_flag='-'*, *sc_dB='-'*, *K_dB='-'*, *pix_area='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Radiometric calibration for multi-look intensity (MLI) data
Copyright 2016, Gamma Remote Sensing, v2.0 9-Nov-2016 uw/clw/of

Parameters

- **MLI** – (input) MLI image (float)
- **MLI_par** – (input) SLC parameter file of input MLI image
- **OFF_par** – (input) ISP offset/interferogram parameter file (enter - for images in MLI geometry)
- **CMLI** – (output) radiometrically calibrated output MLI (float)
- **antenna** – (input) 1-way antenna gain pattern file or - if not provided
- **rloss_flag** –

  range spreading loss correction:

  – 0: no correction (default)
  – 1: apply r^3 correction (all modes except ASAR APS)
  – 2: apply r^4 correction (used only for ASAR APS mode)
  – -1: undo r^3 correction
  – -2: undo r^4 correction)
- **ant_flag** –

  antenna pattern correction:

  – 0: no correction (default)
  – 1: apply antenna pattern correction
  – -1: undo antenna pattern correction)
- **refarea_flag** –

  reference pixel area correction:

  - 0: no pixel area correction (default)
  - 1: calculate sigma0, scale area by sin(inc_ang)/sin(ref_inc_ang)
  - 2: calculate gamma0, scale area by sin(inc_ang)/(cos(inc_ang)*sin(ref_inc_ang)
  - -1: undo sigma0 area scaling factor
  - -2: undo gamma0 area scaling factor
- **sc_dB** – scale factor in dB (default: 0.0)
- **K_dB** – calibration factor in dB (default: -(value from MLI_par))
- **pix_area** – (output) ellipsoid-based ground range sigma0 or gamma0 pixel reference area (float)
- **refarea_flag** – 1 or -1: sigma0 ref. area
- **refarea_flag** – 2 or -2: gamma0 ref. area
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**radcal_PRI**(*PRI*, *PRI_par*, *GRD*, *GRD_par*, *K_dB='-'*, *inc_ref='-'*, *roff='-'*, *nr='-'*, *loff='-'*, *nl='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Convert ESA processed short integer format PRI to radiometrically calibrated GRD image (float)
Copyright 2016, Gamma Remote Sensing, v1.5 5-Mar-2016 uw/clw

Parameters

- **PRI** – (input) PRI ground-range image (short integer, sqrt(backscat. intensity)
- **PRI_par** – (input) SLC parameter file of input PRI ground-range image (yyyymmdd.pri.par)
- **GRD** – (output) calibrated ground-range image (float, backscat. intensity)
- **GRD_par** – (output) ISP image parameter file of output calibrated ground-range image (yyyymmdd.grd.par)
- **K_dB** –

  calibration factor in decibels (default: 59.75 dB)
  ERS1 (D-Paf,ESRIN): 58.24 dB, ERS2 (D-Paf,ESRIN,I-Paf,UK-Paf after 1997): 59.75 dB ENVISAT ASAR: 55.0 dB (all modes) for details see product specifications and ESA publications.
- **inc_ref** –

  reference incidence angle in deg. (default: 23.0 deg.)
  ENVISAT ASAR: 90.0 deg. (all modes)
- **roff** – offset to starting range sample (default: 0)
- **nr** – number of range samples (default: 0, to end of line)
- **loff** – offset to starting line (default: 0, 1 header line in the input file is assumed for ERS)
- **nl** – number of lines to copy (default: 0, to end of file)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in

- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**radcal_SLC**(*SLC*, *SLC_par*, *CSLC*, *CSLC_par*, *fcase='-'*, *antenna='-'*, *rloss_flag='-'*, *ant_flag='-'*, *refarea_flag='-'*, *sc_dB='-'*, *K_dB='-'*, *pix_area='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Radiometric calibration of SLC data
Copyright 2016, Gamma Remote Sensing, v2.4 30-Dec-2017 uw/clw/of

Parameters

- **SLC** – (input) SLC (fcomplex or scomplex)
- **SLC_par** – (input) SLC parameter file of input SLC
- **CSLC** – (output) radiometrically calibrated SLC (fcomplex or scomplex)
- **CSLC_par** – (output) SLC parameter file of output calibrated SLC
- **fcase** –

  format case (default = 1)

  - 1: fcomplex –> fcomplex (pairs of float)
  - 2: fcomplex –> scomplex (pairs of short integer)
  - 3: scomplex –> fcomplex
  - 4: scomplex –> scomplex
- **antenna** – 1-way antenna gain pattern file or - (if not provided)
- **rloss_flag** –

  range spreading loss correction:

  - 0: no correction (default)
  - 1: apply r^3 correction (all modes except ASAR APS)
  - 2: apply r^4 correction (used only for ASAR APS mode)
  - -1: undo r^3 correction
  - -2: undo r^4 correction)
- **ant_flag** –

  antenna pattern correction:

  - 0: no correction (default)
  - 1: apply antenna pattern correction
  - -1: undo antenna pattern correction)
- **refarea_flag** –

  reference pixel area correction:

  - 0: no pixel area correction (default)
  - 1: calculate sigma0, scale area by sin(inc_ang)/sin(ref_inc_ang)
  - 2: calculate gamma0, scale area by sin(inc_ang)/(cos(inc_ang)*sin(ref_inc_ang)
  - -1: undo sigma0 area scaling factor
  - -2: undo gamma0 area scaling factor
- **sc_dB** – scale factor in dB (default: 0.0)

- **K_dB** – calibration factor in dB (default: -(value from SLC_par))
- **pix_area** – (output) ellipsoid-based ground range sigma0 or gamma0 pixel reference area (float)
- **refarea_flag** – 1 or -1: sigma0 ref. area
- **refarea_flag** – 2 or -2: gamma0 ref. area
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**radcal_pwr_stat**(*SLC_tab*, *SLC_tab_cal*, *plist*, *MSR_cal*, *PWR_cal*, *roff='-'*, *loff='-'*, *nr='-'*, *nl='-'*, *plist_out='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate calibrated SLC image files using point targets determined from the Mean/Sigma Ratio and Intensity
Copyright 2018, Gamma Remote Sensing, v1.4 25-Apr-2018 clw/uw/cm

Parameters

- **SLC_tab** – (input) two column list of the SLC filenames and SLC parameter filenames of the uncalibrated SLC images
- **SLC_tab_cal** – (input) two column list of the SLC filenames and SLC parameter filenames of the calibrated SLC images (enter - for none)
- **plist** – (input) point list for the point to use for calibraton (int, enter - to use the data to determine the calibration points)
- **MSR_cal** – mean/sigma ratio for point target selection for relative calibration between scenes: 1.500
- **PWR_cal** – intensity threshold ratio for point target selection for relative calibration between scenes: 1.000
- **roff** – offset to starting range of section to analyze (default -: 0)
- **loff** – offset to starting line of section to analyze (default -: 0)
- **nr** – number of range pixels to analyze (default -: to end of line)
- **nl** – number of azimuth lines to analyze (default -: to end of file)
- **plist_out** – point list of points used to determine calibration using MSR_cal and PWR_cal thresholds
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**rascc_mask**(*cc*, *pwr*, *width*, *start_cc='-'*, *start_pwr='-'*, *nlines='-'*, *pixavr='-'*, *pixavaz='-'*, *cc_thres='-'*, *pwr_thres='-'*, *cc_min='-'*, *cc_max='-'*, *scale='-'*, *exp='-'*, *LR='-'*, *rasf='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Generate phase unwrapping validity mask using correlation and intensity
Copyright 2016, Gamma Remote Sensing, v2.0 12-Sep-2016 clw/uw

Parameters

- **cc** – (input)interferometric correlation image (float)
- **pwr** – (input)intensity image (float, enter - if not available)
- **width** – number of samples/row
- **start_cc** – starting line of coherence image (default: 1)
- **start_pwr** – starting line of intensity image (default: 1)
- **nlines** – number of lines to display (default=0: to end of file)
- **pixavr** – number of pixels to average in range (default: 1)
- **pixavaz** – number of pixels to average in azimuth (default: 1)
- **cc_thres** – coherence threshold for masking, pixels with cc < cc_thres are set to 0 (default: 0.0)
- **pwr_thres** – relative intensity threshold for masking, pixels with intensity < pwr_thres * average intensity are set to 0 (default: 0)
- **cc_min** – minimum coherence value used for color display (default: 0.1)
- **cc_max** – maximum coherence value used for color display (default: 0.9)
- **scale** – intensity display scale factor (default: 1.)
- **exp** – intensity display exponent (default: .35)
- **LR** – left/right mirror image flag, (1: normal (default), -1: mirror image)
- **rasf** –

  (output) image filename, extension determines the format, enter - for default: *.ras *.bmp BMP format *.ras Sun raster format *.tif TIFF format
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**rascc_mask_thinning**(*ras_in*, *in_file*, *width*, *ras_out*, *nmax='-'*, *thresholds='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Adaptive sampling reduction for phase unwrapping validity mask
Copyright 2015, Gamma Remote Sensing, v1.5 5-Dec-2015 uw/clw

Parameters

- **ras_in** – (input) validity mask (SUN/BMP/TIFF raster format 8-bit image)
- **in_file** – (input) file used for adaptive sampling reduction, e.g. correlation coefficient (float)
- **width** – number of samples/row of in_file
- **ras_out** – (output) validity mask with reduced sampling (8-bit SUN rasterfile or BMP format image)
- **nmax** – number of sampling reduction runs (default: 3)
- **thresholds** – a list of thresholds sorted from smallest to largest scale sampling reduction
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in

- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**res_map**(*hgt*, *gr*, *data*, *SLC_par*, *OFF_par*, *res_hgt*, *res_data*, *nr='-'*, *naz='-'*, *azps_res='-'*, *loff='-'*, *nlines='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Slant range to ground range transformation based on interferometric ground-range
Copyright 2008, Gamma Remote Sensing, v2.3 5-Sep-2008 clw/uw

Parameters

- **hgt** – (input) height file in slant range geometry
- **gr** – (input) ground range file in slant range geometry
- **data** – (input) data file in slant range geometry (float) (intensity *.pwr or correlation *.cc)
- **SLC_par** – (input) ISP parameter file of reference SLC
- **OFF_par** – (input) offset/interferogram processing parameters
- **res_hgt** – (output) resampled height file in ground range geometry
- **res_data** – (output) resampled data file in ground range geometry
- **nr** – number of range samples for L.S. estimate (default=7, must be odd)
- **naz** – number of azimuth samples for L.S. extimate (default=7, must be odd)
- **azps_res** – azimuth output map sample spacing in meters (default=azimuth spacing)
- **loff** – offset to starting line for height calculations (default=0)
- **nlines** – number of lines to calculate (default=to end of file)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**residue**(*int*, *flag*, *width*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Determine interferometric phase unwrapping residues
Copyright 2014, Gamma Remote Sensing, v2.6 14-Jan-2014 clw/uw

Parameters

- **int** – (input) interferogram (fcomplex)
- **flag** – (input) flag file (unsigned char)
- **width** – number of samples/row
- **xmin** – offset to starting range pixel(default = 0)
- **xmax** – offset last range pixel (default = width-1)
- **ymin** – offset to starting azimuth row (default = 0)
- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**residue_cc**(*int*, *flag*, *width*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Determine interferometric phase unwrapping residues considering low coherence regions
Copyright 2014, Gamma Remote Sensing, v2.6 20-Jan-2014 clw/uw/ts

Parameters

- **int** – (input) interferogram (fcomplex)
- **flag** – (input) flag file (unsigned char)
- **width** – number of samples/row
- **xmin** – offset to starting range pixel(default = 0)
- **xmax** – offset last range pixel (default = width-1)
- **ymin** – offset to starting azimuth row (default = 0)
- **ymax** – offset to last azimuth row (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**run_all**(*list*, *command*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **list** – (input) list file (ascii)
- **command** –

  command to use
  (example 1: 'ls -l $1.slc') (example 2: 'JERS_PROC JERS-1_720.par $1 2 6 … 8192'
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**sbi_filt**(*SLC_1*, *SLC1_par*, *SLC2R_par*, *SLCf*, *SLCf_par*, *SLCb*, *SLCb_par*, *norm_sq*, *iwflg='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Azimuth filtering of SLC data to support split-beam interferometry to measure azimuth offsets
Copyright 2016, Gamma Remote Sensing, v1.2 clw 5-Mar-2016

Parameters

- **SLC_1** – (input) SLC image (SCOMPLEX or FCOMPLEX format)
- **SLC1_par** – (input) SLC image parameter file
- **SLC2R_par** –

  (input) SLC2 ISP image parameter file for the co-registered image of the interferometric pair,
  used to determine azimuth common-band for each output SLC (enter - for none)

- **SLCf** – (output) SLC image (forward-looking, FCOMPLEX format)
- **SLCf_par** – (output) SLC parameter file (forward-looking)
- **SLCb** – (output) SLC image (backward-looking, FCOMPLEX format)
- **SLCb_par** – (output) SLC parameter file (backward-looking)
- **norm_sq** – squint between beams as a fraction of the azimuth spectrum width (default: 0.5)
- **iwflg** –

  inverse weighting flag:

  - 0: no compensation for azimuth spectrum weighting
  - 1: compensate for the azimuth spectrum weighting (default)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**sbi_offset**(*sbi_unw*, *SLCf_par*, *SLCb_par*, *OFF_par*, *az_offset*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate azimuth offsets from unwrapped split-beam interferogram
Copyright 2011, Gamma Remote Sensing, v1.0 25-Nov-2011

Parameters

- **sbi_unw** – (input) unwrapped phase of split-beam interferogram (float)
- **SLCf_par** – (input) reference SLC parameter file (forward-looking)
- **SLCb_par** – (input) reference SLC parameter file (backward-looking)
- **OFF_par** – (input) offset parameter file
- **az_offset** – (output) azimuth offsets (m)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**slant_range**(*SLC_par*, *slr*, *logpath=None*, *outdir=None*, *shellscript=None*)

Calculate slant range for every range sample
Copyright 2013, Gamma Remote Sensing v1.1 28-Aug-2013

Parameters

- **SLC_par** – (input) SLC or MLI image parameter file
- **slr** – (output) slant range for every sample in the image (float)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**split_WB**(*data_in*, *data_par_in*, *data_tab*, *dtype*, *logpath=None*, *outdir=None*, *shellscript=None*)

ISP: Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/split_WB.c
Copyright 2018, Gamma Remote Sensing, v1.3 25-Apr-2018 clw/cm
Split WB mosaic image into individual beams using ISP parameter files

Parameters

- **data_in** – (input) input mosaicked data in slant-range geometry (e.g. DEM data)
- **data_par_in** – (input) ISP image parameter file for data in the input mosaic
- **data_tab** – (input) 2 column list of output data filenames and ISP image parameter files for each beam in the mosaic (text)
- **dtype** –

  (input) input data type:

  - 0: FLOAT
  - 1: FCOMPLEX
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**subtract_phase**(*interf_in*, *phase_file*, *interf_out*, *width*, *factor='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Land Application Tools: Program /usr/local/GAMMA_SOFTWARE-20180703/ISP/bin/subtract_phase.c
Copyright 2001, Gamma Remote Sensing, v3.1 23-Jan-2001 uw/clw
subtract scaled phase image from a complex interferogram

Parameters

- **interf_in** – (input) input interferogram (fcomplex format)
- **phase_file** – (input) unwrapped interferometric phase (float)
- **interf_out** – (output) output interferogram (input interferogram - scaled phase) (fcomplex)
- **width** – number of samples/line
- **factor** – constant scale factor for input phase data [default=1.0]
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**tree_cc**(*flag*, *width*, *mbl='-'*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Phase unwrapping tree generation with low correlation search (modified ARW algorithm)
Copyright 2014, Gamma Remote Sensing, v2.9 20-Jan-2014 clw/uw

Parameters

- **flag** – (input) phase unwrapping flag file
- **width** – number of samples/row
- **mbl** – maximum branch length (default=32, maximum=64)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row, relative to start (default = 0)
- **ymax** – last azimuth row, relative to start (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**tree_gzw**(*flag*, *width*, *mbl='-'*, *xmin='-'*, *xmax='-'*, *ymin='-'*, *ymax='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Phase unwrapping tree generation (GZW algorithm)
Copyright 2008, Gamma Remote Sensing, v3.6 5-Sep-2008 clw/uw

Parameters

- **flag** – (input) phase unwrapping flag file
- **width** – number of samples/row
- **mbl** – maximum branch length (default=32)
- **xmin** – starting range pixel offset (default = 0)
- **xmax** – last range pixel offset (default = width-1)
- **ymin** – starting azimuth row, relative to start (default = 0)
- **ymax** – last azimuth row, relative to start (default = nlines-1)
- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**unw_correction_filt**(*unw_in*, *unw_out*, *width*, *fsize='-'*, *thresh1='-'*, *thresh2='-'*, *iterations='-'*, *cleaning='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **unw_in** – (input) unwrapped phase file to correct (float)
- **unw_out** – (output) corrected unwrapped phase file (float)
- **width** – number of range samples per line
- **fsize** – maximum filter radius in pixels (default = 5)
- **thresh1** – upper threshold for negative phase differences (default = -3.0)
- **thresh2** – lower threshold for positive phase differences (default = 3.0)
- **iterations** – number of iterations to run (default = 1)

- **cleaning** –

  cleaning flag indicating if intermediary files are deleted (default = 1: yes, 0: no)
  The difference between the unfiltered and spatially filtered phase (using fspf) is used to determine an correct phase unwrapping ambiguity errors

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**unw_correction_poly**(*unw_in*, *unw_out*, *width*, *poly*, *flag*, *max_iter='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Parameters

- **unw_in** – (input) unwrapped phase file to correct (float)
- **unw_out** – (output) corrected unwrapped phase file (float)
- **width** – number of range samples per line
- **poly** – (input) polygon file (text)
- **flag** – ambiguity corrected flag (1: add 2PI; -1: subtract 2PI)
- **max_iter** –

  maximum number of iterations done (default = 1)
  (iterations are used (a) if the ambiguity to correct is not 2PI but a multiple of 2PI and (b) if the ambiguity error is in an area with a significant phase slope)

- **logpath** (*str or None*) – a directory to write command logfiles to
- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

pyroSAR.gamma.parser_demo.**unw_model**(*interf*, *unw_model*, *unw*, *width*, *xinit='-'*, *yinit='-'*, *ref_ph='-'*, *width_model='-'*, *logpath=None*, *outdir=None*, *shellscript=None*)

Phase unwrapping using a model of the unwrapped phase
Copyright 2008, Gamma Remote Sensing, v1.6 5-Sep-2008 clw/uw

Parameters

- **interf** – (input) complex interferogram
- **unw_model** – (input) approximate unwrapped phase model (float)
- **unw** – (output) unwrapped phase (float)
- **width** – number of samples/row of the interferogram
- **xinit** – offset to phase reference location in range (col)
- **yinit** – offset to phase reference location in azimuth (row)
- **ref_ph** – reference point phase (radians) (enter - for phase at the reference point)
- **width_model** – number of samples/row of the unwrapped phase model (default: interferogram width)
- **logpath** (*str or None*) – a directory to write command logfiles to

- **outdir** (*str or None*) – the directory to execute the command in
- **shellscript** (*str or None*) – a file to write the Gamma commands to in shell format

2.4 Sentinel-1 Tools

OSV	interface for management of S1 Orbit State Vector (OSV) files
removeGRDBorderNoise	Mask out Sentinel-1 image border noise.

class pyroSAR.S1.**OSV**(*osvdir=None*, *timeout=300*)

Bases: object

interface for management of S1 Orbit State Vector (OSV) files

input is a directory which is supposed to contain, or already contains, OSV files. Two subdirectories are expected and created otherwise: one for Precise Orbit Ephemerides (POE) named POEORB and one for Restituted Orbit (RES) files named RESORB

Using method *match()* the corresponding POE (priority) or RES file is returned for a timestamp. Timestamps are always handled in the format YYYYmmddTHHMMSS.

Parameters

- **osvdir** (*str*) – the directory to write the orbit files to
- **timeout** (*int or tuple or None*) – the timeout in seconds for downloading OSV files as provided to requests.get()

See also:

requests timeouts

catch(*sensor*, *osvtype='POE'*, *start=None*, *stop=None*, *url_option=1*)

check a server for files

Parameters

- **sensor** (*str or list[str]*) – The S1 mission(s):
  - 'S1A'
  - 'S1B'
  - ['S1A', 'S1B']
- **osvtype** (*str or list[str]*) – the type of orbit files required
- **start** (*str or None*) – the date to start searching for files in format YYYYmmddTHHMMSS
- **stop** (*str or None*) – the date to stop searching for files in format YYYYmmddTHHMMSS
- **url_option** (*int*) – the OSV download URL option
  - 1: https://step.esa.int/auxdata/orbits/Sentinel-1

Returns
the product dictionary of the remote OSV files, with href

Return type
list[dict]

clean_res()

delete all RES files for whose date a POE file exists

date(*file*, *datetype*)

extract a date from an OSV file name

Parameters

- **file** (*str*) – the OSV file
- **datetype** (*{'publish', 'start', 'stop'}*) – one of three possible date types contained in the OSV filename

Returns

a time stamp in the format YYYYmmddTHHMMSS

Return type

str

getLocals(*osvtype='POE'*)

get a list of local files of specific type

Parameters

osvtype (*{'POE', 'RES'}*) – the type of orbit files required

Returns

a selection of local OSV files

Return type

list[str]

match(*sensor*, *timestamp*, *osvtype='POE'*)

return the corresponding OSV file for the provided sensor and time stamp. The file returned is one which covers the acquisition time and, if multiple exist, the one which was published last. In case a list of options is provided as osvtype, the file of higher accuracy (i.e. POE over RES) is returned.

Parameters

- **sensor** (*str*) – The S1 mission:
  - 'S1A'
  - 'S1B'
- **timestamp** (*str*) – the time stamp in the format 'YYYmmddTHHMMSS'
- **osvtype** (*str or list[str]*) – the type of orbit files required; either 'POE', 'RES' or a list of both

Returns

the best matching orbit file (overlapping time plus latest publication date)

Return type

str

maxdate(*osvtype='POE'*, *datetype='stop'*)

return the latest date of locally existing POE/RES files

Parameters

- **osvtype** (*{'POE', 'RES'}*) – the type of orbit files required
- **datetype** (*{'publish', 'start', 'stop'}*) – one of three possible date types contained in the OSV filename

Returns

a timestamp in format YYYYmmddTHHMMSS

Return type

str

mindate(*osvtype='POE'*, *datetype='start'*)

return the earliest date of locally existing POE/RES files

Parameters

- **osvtype** (*{'POE', 'RES'}*) – the type of orbit files required
- **datetype** (*{'publish', 'start', 'stop'}*) – one of three possible date types contained in the OSV filename

Returns

a timestamp in format YYYYmmddTHHMMSS

Return type

str

retrieve(*products*, *pbar=False*)

download a list of product dictionaries into the respective subdirectories, i.e. POEORB or RESORB

Parameters

- **products** (*list[dict]*) – a list of remotely existing OSV product dictionaries as returned by method *catch()*
- **pbar** (*bool*) – add a progressbar?

sortByDate(*files*, *datetype='start'*)

sort a list of OSV files by a specific date type

Parameters

- **files** (*list[str]*) – some OSV files
- **datetype** (*{'publish', 'start', 'stop'}*) – one of three possible date types contained in the OSV filename

Returns

the input OSV files sorted by the defined date

Return type

list[str]

pyroSAR.S1.**removeGRDBorderNoise**(*scene*, *method='pyroSAR'*)

Mask out Sentinel-1 image border noise. This function implements the method for removing GRD border noise as published by ESA [2] and implemented in SNAP and additionally adds further refinement of the result using an image border line simplification approach. In this approach the border between valid and invalid pixels is first simplified using the poly-line vertex reduction method by Visvalingam and Whyatt [4]. The line segments of the new border are then shifted until all pixels considered invalid before the simplification are again on one side of the line. See image below for further clarification.

Parameters

- **scene** (pyroSAR.drivers.SAFE) – the Sentinel-1 scene object
- **method** (*str*) – the border noise removal method to be applied; one of the following:
  - 'ESA': the pure implementation as described by ESA
  - 'pyroSAR': the ESA method plus the custom pyroSAR refinement

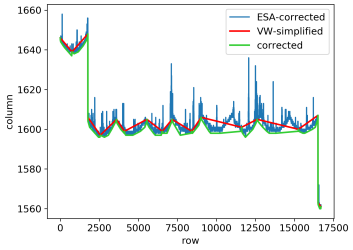

Fig. 4: Demonstration of the border noise removal for a vertical left image border. The area under the respective lines covers pixels considered valid, everything above will be masked out. The blue line is the result of the noise removal as recommended by ESA, in which a lot of noise is still present. The red line is the over-simplified result using the Visvalingam-Whyatt method. The green line is the final result after further correcting the VW-simplified result.

2.5 Auxiliary Data Tools

dem_autoload	obtain all relevant DEM tiles for selected geometries and optionally mosaic them in a VRT.
dem_create	Create a new DEM GeoTIFF file and optionally convert heights from geoid to ellipsoid.
get_egm_lookup	Download lookup tables for converting EGM geoid heights to WGS84 ellipsoid heights.
getasse30_hdr	create an ENVI HDR file for zipped GETASSE30 DEM tiles
get_dem_options	Get the names of all supported DEM type options.
DEMHandler	An interface to obtain DEM data for selected geometries.

class pyroSAR.auxdata.**DEMHandler**(*geometries*)

Bases: object

An interface to obtain DEM data for selected geometries. The files are downloaded into the ESA SNAP auxiliary data directory structure. This class is the foundation for the convenience function *dem_autoload()*.

Parameters
geometries (*list[spatialist.vector.Vector]*) – a list of geometries

property config

static intrange(*extent*, *step*)

generate sequence of integer coordinates marking the tie points of the individual DEM tiles

Parameters

- **extent** (*dict*) – a dictionary with keys *xmin*, *xmax*, *ymin* and *ymax* with coordinates in EPSG:4326.
- **step** (*int*) – the sequence steps

Returns
the integer sequences as (latitude, longitude)

Return type
tuple[range]

load(*dem_type*, *vrt=None*, *buffer=None*, *username=None*, *password=None*, *product='dem'*, *crop=True*, *lock_timeout=600*)

obtain DEM tiles for the given geometries and either return the file names in a list or combine them into a VRT mosaic. The VRT is cropped to the combined extent of the geometries but the pixel grid of the source files is preserved and no resampling/shifting is applied.

Parameters

- **dem_type** (*str*) – the type fo DEM to be used
- **vrt** (*str or None*) – an optional GDAL VRT file created from the obtained DEM tiles
- **buffer** (*int or float or None*) – a buffer in degrees to add around the individual geometries
- **username** (*str or None*) – the download account username
- **password** (*str or None*) – the download account password
- **product** (*str*) –

  the sub-product to extract from the DEM product

  - 'AW3D30'
  - 'dem': the actual Digital Elevation Model
  - 'msk': mask information for each pixel (Cloud/Snow Mask, Land water and low correlation mask, Sea mask, Information of elevation dataset used for the void-filling processing)
  - 'stk': number of DSM-scene files which were used to produce the 5m resolution DSM
  - 'Copernicus 10m EEA DEM'
  - 'dem': the actual Digital Elevation Model
  - 'edm': Editing Mask
  - 'flm': Filling Mask
  - 'hem': Height Error Mask
  - 'wbm': Water Body Mask
  - 'Copernicus 30m Global DEM'
  - 'dem': the actual Digital Elevation Model
  - 'edm': Editing Mask
  - 'flm': Filling Mask
  - 'hem': Height Error Mask
  - 'wbm': Water Body Mask
  - 'Copernicus 30m Global DEM II'
  - 'dem': the actual Digital Elevation Model
  - 'edm': Editing Mask
  - 'flm': Filling Mask
  - 'hem': Height Error Mask
  - 'wbm': Water Body Mask

  - 'Copernicus 90m Global DEM'
  - 'dem': the actual Digital Elevation Model
  - 'edm': Editing Mask
  - 'flm': Filling Mask
  - 'hem': Height Error Mask
  - 'wbm': Water Body Mask
  - 'Copernicus 90m Global DEM II'
  - 'dem': the actual Digital Elevation Model
  - 'edm': Editing Mask
  - 'flm': Filling Mask
  - 'hem': Height Error Mask
  - 'wbm': Water Body Mask
  - 'GETASSE30'
  - 'dem': the actual Digital Elevation Model
  - 'SRTM 1Sec HGT'
  - 'dem': the actual Digital Elevation Model
  - 'SRTM 3Sec'
  - 'dem': the actual Digital Elevation Model
  - 'TDX90m'
  - 'dem': the actual Digital Elevation Model
  - 'am2': Amplitude Mosaic representing the minimum value
  - 'amp': Amplitude Mosaic representing the mean value
  - 'com': Consistency Mask
  - 'cov': Coverage Map
  - 'hem': Height Error Map
  - 'lsm': Layover and Shadow Mask, based on SRTM C-band and Globe DEM data
  - 'wam': Water Indication Mask

- **crop** (*bool*) – If a VRT is created, crop it to spatial extent of the provided geometries or return the full extent of the DEM tiles? In the latter case, the common bounding box of the geometries is expanded so that the coordinates are multiples of the tile size of the respective DEM option.
- **lock_timeout** (*int*) – how long to wait to acquire a lock on downloaded files?

Returns
the names of the obtained files or None if a VRT file was defined

Return type
list[str] or None

remote_ids(*extent*, *dem_type*, *product='dem'*, *username=None*, *password=None*)

parse the names of the remote files overlapping with an area of interest

Parameters

- **extent** (*dict*) – the extent of the area of interest with keys xmin, xmax, ymin, ymax
- **dem_type** (*str*) – the type fo DEM to be used
- **product** (*str*) – the sub-product to extract from the DEM product. Only needed for DEM options 'Copernicus 30m Global DEM' and 'Copernicus 90m Global DEM' and ignored otherwise.
- **username** (*str or None*) – the download account username
- **password** (*str or None*) – the download account password

Returns

the sorted names of the remote files

Return type

str

pyroSAR.auxdata.**dem_autoload**(*geometries*, *demType*, *vrt=None*, *buffer=None*, *username=None*, *password=None*, *product='dem'*, *crop=True*)

obtain all relevant DEM tiles for selected geometries and optionally mosaic them in a VRT.

Parameters

- **geometries** (*list[spatialist.vector.Vector]*) – a list of spatialist.vector.Vector geometries to obtain DEM data for; CRS must be WGS84 LatLon (EPSG 4326)
- **demType** (*str*) – the type of DEM to be used; current options:
  - 'AW3D30' (ALOS Global Digital Surface Model "ALOS World 3D - 30m")
    - info: https://www.eorc.jaxa.jp/ALOS/en/aw3d30/index.htm
    - url: ftp://ftp.eorc.jaxa.jp/pub/ALOS/ext1/AW3D30/release_v1804
    - height reference: EGM96
  - 'Copernicus 10m EEA DEM' (Copernicus 10 m DEM available over EEA-39 countries)
    - registration: https://spacedata.copernicus.eu/web/cscda/data-access/registration
    - url: ftps://cdsdata.copernicus.eu/DEM-datasets/COP-DEM_EEA-10-DGED/2021_1
    - height reference: EGM2008
  - 'Copernicus 30m Global DEM'
    - info: https://copernicus-dem-30m.s3.amazonaws.com/readme.html
    - url: https://copernicus-dem-30m.s3.eu-central-1.amazonaws.com/
    - height reference: EGM2008
  - 'Copernicus 30m Global DEM II'
    - registration: https://spacedata.copernicus.eu/web/cscda/data-access/registration
    - url: ftps://cdsdata.copernicus.eu/DEM-datasets/COP-DEM_GLO-30-DGED/2021_1
    - height reference: EGM2008

  - 'Copernicus 90m Global DEM'
    - info: https://copernicus-dem-90m.s3.amazonaws.com/readme.html
    - url: https://copernicus-dem-90m.s3.eu-central-1.amazonaws.com/
    - height reference: EGM2008
  - 'Copernicus 90m Global DEM II'
    - registration: https://spacedata.copernicus.eu/web/cscda/data-access/registration
    - url: ftps://cdsdata.copernicus.eu/DEM-datasets/COP-DEM_GLO-90-DGED/2021_1
    - height reference: EGM2008
  - 'GETASSE30'
    - info: https://seadas.gsfc.nasa.gov/help-8.1.0/desktop/GETASSE30ElevationModel.html
    - url: https://step.esa.int/auxdata/dem/GETASSE30
    - height reference: WGS84
  - 'SRTM 1Sec HGT'
    - url: https://step.esa.int/auxdata/dem/SRTMGL1
    - height reference: EGM96
  - 'SRTM 3Sec'
    - url: https://download.esa.int/step/auxdata/dem/SRTM90/tiff
    - height reference: EGM96
  - 'TDX90m'
    - registration: https://geoservice.dlr.de/web/dataguide/tdm90
    - url: ftpes://tandemx-90m.dlr.de
    - height reference: WGS84
- **vrt** (*str or None*) – an optional GDAL VRT file created from the obtained DEM tiles
- **buffer** (*int, float, None*) – a buffer in degrees to add around the individual geometries
- **username** (*str or None*) – (optional) the username for services requiring registration
- **password** (*str or None*) – (optional) the password for the registration account
- **product** (*str*) – the sub-product to extract from the DEM product. The following options are available for the respective DEM types:
  - 'AW3D30'
    - 'dem': the actual Digital Elevation Model
    - 'msk': mask information for each pixel (Cloud/Snow Mask, Land water and low correlation mask, Sea mask, Information of elevation dataset used for the void-filling processing)
    - 'stk': number of DSM-scene files which were used to produce the 5 m resolution DSM
  - 'Copernicus 10m EEA DEM'

    - 'dem': the actual Digital Elevation Model
    - 'edm': editing mask
    - 'flm': filling mask
    - 'hem': height error mask
    - 'wbm': water body mask
  - 'Copernicus 30m Global DEM'
    - 'dem': the actual Digital Elevation Model
    - 'edm': Editing Mask
    - 'flm': Filling Mask
    - 'hem': Height Error Mask
    - 'wbm': Water Body Mask
  - 'Copernicus 30m Global DEM II'
    - 'dem': the actual Digital Elevation Model
    - 'edm': editing mask
    - 'flm': filling mask
    - 'hem': height error mask
    - 'wbm': water body mask
  - 'Copernicus 90m Global DEM'
    - 'dem': the actual Digital Elevation Model
    - 'edm': Editing Mask
    - 'flm': Filling Mask
    - 'hem': Height Error Mask
    - 'wbm': Water Body Mask
  - 'Copernicus 90m Global DEM II'
    - 'dem': the actual Digital Elevation Model
    - 'edm': editing mask
    - 'flm': filling mask
    - 'hem': height error mask
    - 'wbm': water body mask
  - 'GETASSE30'
    - 'dem': the actual Digital Elevation Model
  - 'SRTM 1Sec HGT'
    - 'dem': the actual Digital Elevation Model
  - 'SRTM 3Sec'
    - 'dem': the actual Digital Elevation Model
  - 'TDX90m'
    - 'dem': the actual Digital Elevation Model
    - 'am2': Amplitude Mosaic representing the minimum value
    - 'amp': Amplitude Mosaic representing the mean value

* 'com': Consistency Mask
* 'cov': Coverage Map
* 'hem': Height Error Map
* 'lsm': Layover and Shadow Mask, based on SRTM C-band and Globe DEM data
* 'wam': Water Indication Mask

- **crop** (*bool*) – crop to the provided geometries (or return the full extent of the DEM tiles)?

Returns
the names of the obtained files or None if a VRT file was defined

Return type
list[str] or None

Examples

download all SRTM 1 arcsec DEMs overlapping with a Sentinel-1 scene and mosaic them to a single GeoTIFF file

```
from pyroSAR import identify
from pyroSAR.auxdata import dem_autoload
from spatialist import gdalwarp

# identify the SAR scene
filename = 'S1A_IW_SLC__1SDV_20150330T170734_20150330T170801_005264_006A6C_DA69.
↪zip'
scene = identify(filename)

# extract the bounding box as spatialist.Vector object
bbox = scene.bbox()

# download the tiles and virtually combine them in an in-memory
# VRT file subsetted to the extent of the SAR scene plus a buffer of 0.01 degrees
vrt = '/vsimem/srtm1.vrt'
dem_autoload(geometries=[bbox], demType='SRTM 1Sec HGT',
             vrt=vrt, buffer=0.01)

# write the final GeoTIFF file
outname = scene.outname_base() + 'srtm1.tif'
gdalwarp(src=vrt, dst=outname, options={'format': 'GTiff'})

# alternatively use function dem_create and warp the DEM to UTM
# including conversion from geoid to ellipsoid heights
from pyroSAR.auxdata import dem_create
outname = scene.outname_base() + 'srtm1_ellp.tif'
dem_create(src=vrt, dst=outname, t_srs=32632, tr=(30, 30),
           geoid_convert=True, geoid='EGM96')
```

pyroSAR.auxdata.**dem_create**(*src*, *dst*, *t_srs=None*, *tr=None*, *threads=None*, *geoid_convert=False*, *geoid='EGM96'*, *nodata=None*, *resampleAlg='bilinear'*, *dtype=None*, *pbar=False*, *lock_timeout=600*, ***kwargs*)

Create a new DEM GeoTIFF file and optionally convert heights from geoid to ellipsoid. This is basically a convenience wrapper around `osgeo.gdal.Warp()` via `spatialist.auxil.gdalwarp()`. The following argument defaults deviate from those of `osgeo.gdal.WarpOptions()`:

- *format* is set to 'GTiff'
- *resampleAlg* is set to 'bilinear'
- *targetAlignedPixels* is set to 'True'

Parameters

- **src** (*str*) – the input dataset, e.g. a VRT from function *dem_autoload()*
- **dst** (*str*) – the output dataset
- **t_srs** (*None, int, str or osgeo.osr.SpatialReference*) – A target geographic reference system in WKT, EPSG, PROJ4 or OPENGIS format. See function spatialist.auxil.crsConvert() for details. Default (None): use the crs of src.
- **tr** (*None or tuple[int or float]*) – the target resolution as (xres, yres)
- **threads** (*int, str or None*) – the number of threads to use. Possible values:
  - Default *None*: use the value of *GDAL_NUM_THREADS* without modification. If *GDAL_NUM_THREADS* is None, multi-threading is still turned on and two threads are used, one for I/O and one for computation.
  - integer value: temporarily modify *GDAL_NUM_THREADS* and reset it once done. If 1, multithreading is turned off.
  - *ALL_CPUS*: special string to use all cores/CPUs of the computer; will also temporarily modify *GDAL_NUM_THREADS*.
- **geoid_convert** (*bool*) – convert geoid heights?
- **geoid** (*str*) – the geoid model to be corrected, only used if geoid_convert == True; current options:
  - 'EGM96'
  - 'EGM2008'
- **nodata** (*int or float or str or None*) – the no data value of the source and destination files. Can be used if no source nodata value can be read or to override it. A special string 'None' can be used to skip reading the value from the source file.
- **resampleAlg** (*str*) – the resampling algorithm tu be used. See here for options: https://gdal.org/programs/gdalwarp.html#cmdoption-gdalwarp-r
- **dtype** (*str or None*) – override the data type of the written file; Default None: use same type as source data. Data type notations of GDAL (e.g. *Float32*) and numpy (e.g. *int8*) are supported. See spatialist.raster.Dtype.
- **pbar** (*bool*) – add a progressbar?
- **lock_timeout** (*int*) – how long to wait to acquire a lock on *dst*?
- ****kwargs** – additional keyword arguments to be passed to spatialist.auxil.gdalwarp(). See osgeo.gdal.WarpOptions() for options. The following arguments cannot be set as they are controlled internally:
  - *xRes*, *yRes*: controlled via argument *tr*
  - *srcSRS*, *dstSRS*: controlled via the CRS of *src* and arguments *t_srs*, *geoid*, *geoid_convert*
  - *srcNodata*, *dstNodata*: controlled via argument *nodata*
  - *outputType*: controlled via argument *dtype*
  - *multithread* controlled via argument *threads*

pyroSAR.auxdata.**get_dem_options**(*require_auth=None*)

Get the names of all supported DEM type options.

Parameters

require_auth (*bool or None*) – only return options that do/don't require authentication. Default None: return all options.

Returns

the names of the DEM options

Return type

list[str]

pyroSAR.auxdata.**get_egm_lookup**(*geoid*, *software*)

Download lookup tables for converting EGM geoid heights to WGS84 ellipsoid heights.

Parameters

- **geoid** (*str*) – the geoid model; current options:
  - SNAP: 'EGM96'
  - PROJ: 'EGM96', 'EGM2008'
- **software** (*str*) – the software for which to download the EGM lookup
  - SNAP: default directory: `~/.snap/auxdata/dem/egm96`; URL:
    - https://step.esa.int/auxdata/dem/egm96/ww15mgh_b.zip
  - PROJ: requires `PROJ_DATA` or `PROJ_LIB` environment variable to be set as download directory; URLs:
    - https://cdn.proj.org/us_nga_egm96_15.tif
    - https://cdn.proj.org/us_nga_egm08_25.tif

pyroSAR.auxdata.**getasse30_hdr**(*fname*)

create an ENVI HDR file for zipped GETASSE30 DEM tiles

Parameters

fname (*str*) – the name of the zipped tile

2.6 Datacube Tools

This (still experimental) module is intended to easily prepare SAR scenes processed by pyroSAR for ingestion into an Open Data Cube.

```
from pyroSAR.datacube_util import Product, Dataset
from pyroSAR.ancillary import find_datasets

# find pyroSAR files by metadata attributes
archive_s1 = '/.../sentinel1/GRD/processed'
scenes_s1 = find_datasets(archive_s1, sensor=('S1A', 'S1B'), acquisition_mode='IW')

# group the found files by their file basenames
# files with the same basename are considered to belong to the same dataset
grouped = groupby(scenes_s1, 'outname_base')

# define the polarization units describing the data sets
units = {'VV': 'backscatter VV', 'VH': 'backscatter VH'}

# create a new product
```

(continues on next page)

(continued from previous page)

```
with Product(name='S1_GRD_index',
             product_type='gamma0',
             description='Gamma Naught RTC backscatter') as prod:

    for dataset in grouped:
        with Dataset(dataset, units=units) as ds:

            # add the dataset to the product
            prod.add(ds)

            # parse datacube indexing YMLs from product and data set metadata
            prod.export_indexing_yml(ds, 'yml_index_outdir')

    # write the product YML
    prod.write('yml_product')

    # print the product metadata which is written to the product YML
    print(prod)
```

class pyroSAR.datacube_util.**Dataset**(*filename*, *units='DN'*)

Bases: object

A general class describing dataset information required for creating ODC YML files

Parameters

- **filename** (*str*, *list*, Dataset) – the product to be used; either an existing *Dataset* object or a (list of) file(s) matching the pyroSAR naming pattern, i.e. that can be parsed by *pyroSAR.ancillary.parse_datasetname()*
- **units** (*str or dict*) – the units of the product measurement

__add__(*dataset*)

override the + operator. This is intended to easily combine two Dataset objects, which were created from different files belonging to the same measurement, e.g. two GeoTIFFs with one polarization each.

Parameters
dataset (Dataset) – the dataset to add to the current one

Returns
the combination of the two

Return type
Dataset

__radd__(*dataset*)

similar to *Dataset.__add__()* but for function sum(), e.g. sum([Dataset1, Dataset2])

Parameters
dataset (Dataset) – the dataset to add to the current one

Returns
the combination of the two

Return type
Dataset

close()

property filenames

Returns
all file names registered in the dataset

Return type
dict

property **identifier**

Returns
a unique dataset identifier

Return type
str

property **units**

Returns
all measurement unit names registered in the dataset

Return type
dict

class pyroSAR.datacube_util.**Product**(*definition=None*, *name=None*, *product_type=None*, *description=None*)

Bases: object

A class for describing an ODC product definition

Parameters

- **definition** (*str*, *list*, *None*) – the source of the product definition; either an existing product YML, a list of *Dataset* objects, or None. In the latter case the product is defined using the parameters *name*, *product_type* and *description*.
- **name** (*str*) – the name of the product in the data cube
- **product_type** (*str*) – the type of measurement defined in the product, e.g. *gamma0*
- **description** (*str*) – the description of the product and its measurements

add(*dataset*)

Add a dataset to the abstracted product description. This first performs a check whether the dataset is compatible with the product and its already existing measurements. If a measurement in the dataset does not yet exist in the product description it is added.

Parameters
dataset (Dataset) – the dataset whose description is to be added

check_integrity(*dataset*, *allow_new_measurements=False*)

check if a dataset is compatible with the product definition.

Parameters

- **dataset** (Dataset) – the dataset to be checked
- **allow_new_measurements** (*bool*) – allow new measurements to be added to the product definition? If not and the dataset contains measurements, which are not defined in the product, an error is raised.

Raises
RuntimeError –

close()

export_indexing_yml(*dataset*, *outdir*)

Write a YML file named {*Dataset.identifier()*}_dcindex.yml, which can be used for indexing a dataset in an Open Data Cube. The file will contain information from the product and the dataset and a test is first performed to check whether the dataset matches the product definition. A unique ID is issued using uuid.uuid4().

Parameters

- **dataset** (Dataset) – the dataset for which to export a file for
- **outdir** (*str*) – the directory to write the file to

export_ingestion_yml(*outname*, *product_name*, *ingest_location*, *chunking*)

Write a YML file, which can be used for ingesting indexed datasets into an Open Data Cube.

Parameters

- **outname** (*str*) – the name of the YML file to write
- **product_name** (*str*) – the name of the product in the ODC
- **ingest_location** (*str*) – the location of the ingested NetCDF files
- **chunking** (*dict*) – a dictionary with keys 'x', 'y' and 'time'; determines the size of the netCDF files ingested into the datacube; e.g. {'x': 512, 'y': 512, 'time': 1}

property measurements

Returns
a dictionary with measurement names as keys

Return type
dict of dict

write(*ymlfile*)

write the product definition to a YML file

Parameters
ymlfile (*str*) – the file to write to

2.7 Ancillary Functions

This module gathers central functions and classes for general pyroSAR applications.

find_datasets	find pyroSAR datasets in a directory based on their metadata
getargs	get the arguments of a function
groupby	group a list of images by a metadata attribute
groupbyTime	function to group images by their acquisition time difference
hasarg	simple check whether a function takes a parameter as input
multilook_factors	compute multi-looking factors to approximate a square pixel with defined target ground range pixel spacing.
parse_datasetname	Parse the name of a pyroSAR processing product and extract its metadata components as dictionary
seconds	function to extract time in seconds from a file name.
Lock	File and folder locking mechanism.
LockCollection	Like *Lock* but for multiple files/folders.

class pyroSAR.ancillary.**Lock**(*target*, *soft=False*, *timeout=7200*)

Bases: object

File and folder locking mechanism. This mechanism creates lock files indicating whether a file/folder

1. is being modified (*target*.lock),
2. is being used/read (*target*.used_<uuid.uuid4>) or
3. was damaged during modification (*target*.error).

Although these files will not prevent locking by other mechanisms (UNIX locks are generally only advisory), this mechanism is respected across any running instances. I.e., if such a lock file exists, no process trying to acquire a lock using this class will succeed if a lock file intending to prevent it exists. This was implemented because other existing solutions like filelock or fcntl do not implement effective solutions for parallel jobs in HPC systems.

Hard locks prevent any usage of the data. Damage/error locks work like hard locks except that *timeout* is ignored and a *RuntimeError* is raised immediately. Error locks are created if an error occurs whilst a hard lock is acquired and *target* exists (by renaming the hard lock file). Infinite usage locks may exist, each with a different random UUID. No hard lock may be acquired whilst usage locks exist. On error usage locks are simply deleted.

It may happen that lock files remain when a process is killed by HPC schedulers like Slurm because in this case the process is not ended by Python. Optimally, hard locks should be renamed to error lock files and usage lock files should be deleted. This has to be done separately.

Examples

```
>>> from pyroSAR.ancillary import Lock
>>> target = 'test.txt'
>>> with Lock(target=target):
>>>     with open(target, 'w') as f:
>>>         f.write('Hello World!')
```

Parameters

- **target** (*str*) – the file/folder to lock
- **soft** (*bool*) – lock the file/folder only for reading (and not for modification)?
- **timeout** (*int*) – the time in seconds to retry acquiring a lock

is_used()

Does any usage lock exist?

Return type
bool

remove()

Remove the acquired soft/hard lock

class pyroSAR.ancillary.**LockCollection**(*targets*, *soft=False*, *timeout=7200*)

Bases: object

Like *Lock* but for multiple files/folders.

Parameters

- **targets** (*list[str]*) – the files/folders to lock
- **soft** (*bool*) – lock the files/folders only for reading (and not for modification)?
- **timeout** (*int*) – the time in seconds to retry acquiring a lock

pyroSAR.ancillary.**find_datasets**(*directory*, *recursive=False*, ***kwargs*)

find pyroSAR datasets in a directory based on their metadata

Parameters

- **directory** (*str*) – the name of the directory to be searched
- **recursive** (*bool*) – search the directory recursively into subdirectories?
- **kwargs** – Metadata attributes for filtering the scene list supplied as *key=value*. e.g. *sensor='S1A'*. Multiple allowed options can be provided in tuples, e.g. *sensor=('S1A', 'S1B')*. Any types other than tuples require an exact match, e.g. *proc_steps=['grd', 'mli', 'geo', 'norm', 'db']* will be matched only if these processing steps are contained in the product name in this exact order. The special attributes *start* and *stop* can be used for time filtering where *start<=value<=stop*. See function *parse_datasetname()* for further options.

Returns

the file names found in the directory and filtered by metadata attributes

Return type

list of str

Examples

```
>>> selection = find_datasets('path/to/files', sensor=('S1A', 'S1B'),␣
↪polarization='VV')
```

pyroSAR.ancillary.**getargs**(*func*)

get the arguments of a function

Parameters

func (*function*) – the function to be checked

Returns

the argument names

Return type

list or str

pyroSAR.ancillary.**groupby**(*images*, *attribute*)

group a list of images by a metadata attribute

Parameters

- **images** (*list[str]*) – the names of the images to be sorted
- **attribute** (*str*) – the name of the attribute used for sorting; see *parse_datasetname()* for options

Returns

a list of sub-lists containing the grouped images

Return type

list[list[str]]

pyroSAR.ancillary.**groupbyTime**(*images*, *function*, *time*)

function to group images by their acquisition time difference

Parameters

- **images** (*list[str]*) – a list of image names
- **function** (*function*) – a function to derive the time from the image names; see e.g. *seconds()*

- **time** (*int or float*) – a time difference in seconds by which to group the images

Returns
a list of sub-lists containing the grouped images

Return type
list[list[str]]

pyroSAR.ancillary.**hasarg**(*func*, *arg*)
simple check whether a function takes a parameter as input

Parameters

- **func** (*function*) – the function to be checked
- **arg** (*str*) – the argument name to be found

Returns
does the function take this as argument?

Return type
bool

pyroSAR.ancillary.**multilook_factors**(*source_rg*, *source_az*, *target*, *geometry*, *incidence*)
compute multi-looking factors to approximate a square pixel with defined target ground range pixel spacing.

Parameters

- **source_rg** (*int or float*) – the range pixel spacing
- **source_az** (*int or float*) – the azimuth pixel spacing
- **target** (*int or float*) – the target pixel spacing of an approximately square pixel
- **geometry** (*str*) – the imaging geometry; either 'SLANT_RANGE' or 'GROUND_RANGE'
- **incidence** (*int or float*) – the angle of incidence

Returns
the multi-looking factors as (range looks, azimuth looks)

Return type
tuple[int]

Examples

```
>>> from pyroSAR.ancillary import multilook_factors
>>> rlks, azlks = multilook_factors(source_rg=2, source_az=13, target=10,
>>>                                 geometry='SLANT_RANGE', incidence=39)
>>> print(rlks, azlks)
4 1
```

pyroSAR.ancillary.**parse_datasetname**(*name*, *parse_date=False*)
Parse the name of a pyroSAR processing product and extract its metadata components as dictionary

Parameters

- **name** (*str*) – the name of the file to be parsed
- **parse_date** (*bool*) – parse the start date to a datetime object or just return the string?

Returns
the metadata attributes

Return type
dict

Examples

```
>>> meta = parse_datasetname('S1A__IW___A_20150309T173017_VV_grd_mli_geo_norm_db.
↪tif')
>>> print(sorted(meta.keys()))
['acquisition_mode', 'extensions', 'filename', 'orbit',
'outname_base', 'polarization', 'proc_steps', 'sensor', 'start']
```

pyroSAR.ancillary.**seconds**(*filename*)

function to extract time in seconds from a file name. the format must follow a fixed pattern: YYYYmmddTHHMMSS Images processed with pyroSAR functionalities via module snap or gamma will contain this information.

Parameters
filename (*str*) – the name of a file from which to extract the time from

Returns
the difference between the time stamp in filename and Jan 01 1900 in seconds

Return type
float

pyroSAR.ancillary.**windows_fileprefix**(*func*, *path*, *exc_info*)

Helper function for `shutil.rmtree()` to exceed Windows' file name length limit of 256 characters. See here for details.

Parameters

- **func** (*function*) – the function to be executed, i.e. `shutil.rmtree()`
- **path** (*str*) – the path to be deleted
- **exc_info** (*tuple*) – execution info as returned by `sys.exc_info()`

Examples

```
>>> import shutil
>>> from pyroSAR.ancillary import windows_fileprefix
>>> shutil.rmtree('/path', onerror=windows_fileprefix)
```

2.8 Examine

class pyroSAR.examine.**ExamineGamma**

Bases: `object`

Class to check if GAMMA is installed.

Examples

```
>>> from pyroSAR.examine import ExamineGamma
>>> config = ExamineGamma()
>>> print(config.home)
>>> print(config.version)
```

class pyroSAR.examine.**ExamineSnap**

Bases: object

Class to check if ESA SNAP is installed. Upon initialization, this class searches for relevant binaries and the accompanying relative directory structure, which uniquely identify an ESA SNAP installation on a system. First, all relevant file and folder names are read from the pyroSAR config file if it exists and their existence is verified. If this fails, a system check is performed to find relevant binaries in the system PATH variable and additional files and folders relative to them. Furthermore, a snap.auxdata.properties file is scanned for auxiliary data URLs and local storage location. This is used by SNAP to manage data from e.g. the SRTM mission. In case SNAP is not installed, the respective information is read from a default file delivered with pyroSAR. This has the advantage of using the SNAP download URLs and local directory structure without having SNAP installed such that it can be adapted by other SAR software.

get_suffix(*operator*)

get the file name suffix for an operator

Parameters
operator (*str*) – the name of the operator

Returns
the file suffix

Return type
str

Examples

```
>>> from pyroSAR.examine import ExamineSnap
>>> config = ExamineSnap()
>>> print(config.get_suffix('Terrain-Flattening'))
'TF'
```

get_version(*module*)

Read the version and date of different SNAP modules. This scans a file 'messages.log', which is re-written every time SNAP is started.

Parameters
module (*str*) – one of the following

- core
- desktop
- rstbx
- s1tbx
- s2tbx
- s3tbx

Returns
a dictionary with keys 'version' and 'date'

Return type
dict

CHAPTER

THREE

ABOUT

3.1 Projects using pyroSAR

pyroSAR is/was used in these projects:

- BACI
- CCI Biomass
- COPA
- EMSAfrica
- GlobBiomass
- SALDi
- SenThIS
- Sentinel4REDD
- SWOS
- BONDS

You know of other projects? We'd be happy to know.

3.2 Changelog

3.2.1 0.6 | 2018-11-20

SAR metadata

- new standardized metadata fields *orbitNumber_abs*, *orbitNumber_rel*, *cycleNumber* and *frameNumber* for all SAR formats
- customization of output file names with additional metadata fields (e.g. orbit numbers)

software configuration

- pyroSAR configuration file handling: the paths to the SNAP and Gamma installation as well as relevant metadata directories are now registered in a configuration file *config.ini*, which is stored in a directory *.pyrosar* in the user home directory
- improved SNAP installation verification: pyroSAR now performs a deeper check of the SNAP installation to make sure it is not mistaken with e.g. the Ubuntu package manager snap; relevant installation executables and directories are stored in the configuration file

general functionality

- deeper integration of package spatialist: all the spatial file handling functionality that was part of pyroSAR is now part of package spatialist; now all the functionality is imported from spatialist and removed from pyroSAR
- improved search for datasets processed by pyroSAR: new helper functions exist, which make it easier to search for datasets by metadata fields, which are internally searched for in the respective file names
- introduced gamma function parser: these new tools search for a GAMMA_HOME environment variable and, if found, parse Python functions from the docstring of respective command line tools; for this, new Python scripts are created, which are stored alongside the configuration file in the user home directory; this way users can easily use Python functions with named parameters instead of the positional arguments of the Gamma command line tools
- improved documentation

Open Data Cube Export

functionality to export processed datasets directly to an Open Data Cube: it is now possible to create Open Data Cube product YML files as well as YML files for data indexing and ingestion into this product; pyroSAR also internally checks for compatibility of a particular dataset with the target product; this way, the resulting files can easily be passed to the Open Data Cube command line tools several bug fixes

SNAP API

improved SNAP processing workflow node linking: it is now possible to add a node also before an existing one, instead of just after it

Python package integrity

- add trove classifiers for supported operating systems and MIT license for easier online search
- exchange http with https for all URLs that support it

3.2.2 0.7 | 2019-01-03

several changes to the functioning of the Gamma command API

GAMMA API

processing

- *pyroSAR.gamma.geocode()*:
  - optionally write all Gamma commands to shellscript
  - newly introduced choice of normalization method
  - changed normalization default approach
- *pyroSAR.gamma.process()*:
  - new parameter *logfile* to specify a logfile instead of just a directory with automated file naming
  - new parameter *shellscript* to write the executed command to a shell script protocol

command parser

- add parameters *outdir* and *shellscript* to parsed functions
- extensive improvement to accurately parse more commands
- add parameter *inlist* to some commands, which require interactive input via *stdin*

general

- several bug fixes
- extended documentation
- make use of parsed command functions internally
- enable passing *logpath*, *outdir* and *shellscript* to all parsed functions via additional parameters for other convenience functions

3.2.3 0.8 | 2019-02-11

Auxiliary Data Handling

- new module auxdata with function *pyroSAR.auxdata.dem_autoload()* to automatically download tiles of different DEM types overlapping with given geometries
- class *pyroSAR.S1.OSV*: reduced search time for new RES orbit state vector files; included more meaningful status messages

GAMMA API

- new function `pyroSAR.gamma.srtm.dem_autocreate()` to automatically create DEMs in Gamma format from the output of function *pyroSAR.auxdata.dem_autoload()*
- improved writing of ENVI HDR files from class *pyroSAR.gamma.ISPPar*
- class *pyroSAR.gamma.UTM*: improved to work with newer Gamma versions
- function *pyroSAR.gamma.geocode()*:
  - improved documentation
  - clarified code for better readability
  - more consistent naming scheme for all temporarily written files

  - export temporarily written files (e.g. local incidence angle) via new parameter *export_extra*
  - additional parametrization tests to ensure best processing result
  - changed default of parameter *func_interp* to 2 to work best with default of parameter *normalization_method* (see documentation of Gamma command pixel_area)

SNAP API

- function *pyroSAR.snap.util.geocode()*:
  - export temporarily written files (e.g. local incidence angle) via new parameter *export_extra*

3.2.4 0.9 | 2019-06-15

Drivers

- *pyroSAR.drivers.SAFE*: read heading angle, incident angle and image geometry (e.g. Ground Range) from metadata
- *pyroSAR.drivers.Archive*: improved cross-compatibility with Python2 and Python3

SNAP API

- function *pyroSAR.snap.util.geocode()*:
  - option to export *DEM* via parameter *export_extra*
  - added Sentinel-1 *ThermalNoiseRemoval* node via new parameter *removeS1ThermalNoise*
  - added *Multilook* node which is executed to approximate the target resolution if necessary (currently only for Sentinel-1 since metadata entries *incidence* and *image_geometry* are required)
  - new parameter *groupsize* to split workflows into several groups, which are executed separately with intermediate products written to disk. This increases processing speed
  - simplified internal node parametrization for easier use in future functions
  - fail if no POE orbit state vector file is found
  - *Terrain-Flattening*:
    - added additional parameters *additionalOverlap* and *oversamplingMultiple*
    - use bilinear instead of bicubic interpolation
  - *Remove-GRD-Border-Noise*: decrease *borderLimit* from 1000 to 500 (SNAP default)
  - new parameter *gpt_exceptions* to execute workflows containing specific nodes with different GPT versions than the default one
  - automatically remove node parameters on GPT fail and re-run the modified workflow; this is relevant if a node is executed in an older GPT version (e.g. via parameter *gpt_exceptions*), which does not accept parameters which were introduced in later GPT versions (e.g. those described above for node *Terrain-Flattening*)
  - disable/enable terrain flattening via new parameter *terrainFlattening*
  - optionally return workflow filename with new parameter *returnWF*
  - execute custom pyroSAR S1 GRD border noise removal (see *pyroSAR.S1.removeGRDBorderNoise()*)
  - new parameters *demResamplingMethod* and *imgResamplingMethod*

GAMMA API

- SRTM Tools renamed to DEM Tools
  - function *pyroSAR.gamma.dem.dem_autocreate()*:
    - define arbitrary output CRS and resolution via new parameters *t_srs* and *tr*
    - optionally perform geoid to ellipsoid conversion in either GDAL or GAMMA via new parameter *geoid_mode*
- function *pyroSAR.gamma.geocode()*:
  - removed multiplication of backscatter with cosine of incident angle via command *lin_comb*
  - fixed bug in writing correct nodata values to ancillary products defined via parameter *export_extra*
  - changed default of parameter *func_geoback* from 2 to 1 (GAMMA default)
- function *pyroSAR.gamma.correctOSV()*:
  - fixed bug in using the first OSV file in a directory for correcting an image, which resulted in S1B files being corrected with S1A OSV files. This occasionally resulted in errors of no DEM overlap while processing S1B scenes
- fixed bug in treating GAMMA image pixel coordinates as top left instead of pixel center. This is relevant for writing ENVI HDR files for GAMMA images via function *pyroSAR.gamma.par2hdr()* resulting in the image to be shifted by 1/2 pixel to Southeast

Command Parser

- compatibility with GAMMA version released in November 2018
- delete parsed modules if environment variable *GAMMA_HOME* was reset causing them to be re-parsed with the new version on module import

general functionality

- new function *pyroSAR.ancillary.multilook_factors()* to compute factors depending on image geometry and target resolution
- *pyroSAR.S1.removeGRDBorderNoise()*: reached Python3 compatibility

Auxiliary Data Handling

- new function *pyroSAR.auxdata.dem_create()* for convenient creation of DEM mosaics as downloaded by *pyroSAR.auxdata.dem_autoload()*
- function *pyroSAR.auxdata.dem_autoload()*: download 1 degree tiles instead of 5 degree tiles
- class *pyroSAR.S1.OSV*:
  - download files specific to the Sentinel-1 sensor (S1A/S1B) instead of all matching the acquisition time
  - improved time span search, which occasionally resulted in missing OSV files

3.2.5 0.9.1 | 2019-07-05

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_create()*: new parameter *resampling_method*

GAMMA API

- function *pyroSAR.gamma.dem.dem_autocreate()*: new parameter *resampling_method*

SNAP API

- function *pyroSAR.snap.util.geocode()*: fixed typo of parameter *removeS1BorderNoise*

3.2.6 0.10 | 2019-12-06

Drivers

- method *bbox()*: choose the output vector file format via new parameter *driver* or by using one of spatialist's supported file name extensions (see `spatialist.vector.Vector.write()`)
- *pyroSAR.drivers.SAFE*
  - new method *quicklook()* for writing KMZ quicklooks
  - method *getOSV()*: renamed parameter *outdir* to *osvdir*
- *pyroSAR.drivers.Archive*: remove scenes from the database if they cannot be found at their file location. This is performed at each initialization of an *Archive* object.

GAMMA API

- new parameter *basename_extensions* for adding extra metadata fields to output image names; affects:
  - *pyroSAR.gamma.convert2gamma()*
  - *pyroSAR.gamma.geocode()*
- *pyroSAR.gamma.correctOSV()*: make use of OSV files in SNAP's auxdata structure
- *pyroSAR.gamma.geocode()*: made border nose removal optional with new parameter *removeS1BorderNoise*

SNAP API

- workflow parsing
  - improved output XML for better display in SNAP GUI
  - support for nodes with multiple input scenes, e.g. *SliceAssembly*
- SAR processor (function *gpt()*)
  - write Sentinel-1 manifest.safe with processing results
  - two methods for border noise removal: *ESA* and *pyroSAR* via new parameter *removeS1BorderNoiseMethod*
- function *pyroSAR.snap.util.geocode()*
  - optional speckle filtering with new parameter *speckleFilter*

  - choose the output backscatter reference area (*beta0*/*gamma0*/*sigma0*) with new parameter *refarea*
  - default of parameter *groupsize* changed to 1
  - internally download S1 OSV files
  - internally download SNAP's *EGM96* geoid to *WGS84* ellipsoid DEM conversion lookup table via new function `pyroSAR.snap.auxil.get_egm96_lookup()`
  - support for multi-scene *SliceAssembly*; can be invoke by passing a list of scenes to parameter *infile*
  - new parameter *removeS1BorderNoiseMethod*
  - new parameter *gpt_args* to pass additional arguments to the GPT call

Datacube Tools

- *pyroSAR.datacube_util.Product.export_ingestion_yml()*: new parameter *chunking*

Auxiliary Data Handling

- OSV download functionality (class *pyroSAR.S1.OSV*)
  - made definition of OSV download directory optional; default is SNAP's auxdata directory
  - organization of downloaded files into SNAP's auxdata structure:
    * compression to zip
    * sort files into subdirs for sensor, year, month
  - removed method `update()`

Ancillary Tools

- *pyroSAR.ancillary.parse_datasetname()*
  - support for datasets in NetCDF format
  - enable parsing of ancillary products like local incidence angle (*inc_geo.tif)
- *pyroSAR.ancillary.find_datasets()*: new parameters *start* and *stop* for time filtering

general

- bug fixes and documentation improvements

3.2.7 0.10.1 | 2019-12-12

GAMMA API

- *Command API* compatibility with GAMMA version 20191203

3.2.8 0.11 | 2020-05-29

Drivers

- *pyroSAR.drivers.Archive*: completely restructured to use the SQLAlchemy Object Relational Mapper (ORM). This makes it possible to switch between SQLite+Spatialite and PostgreSQL+PostGIS database backends.
- *pyroSAR.drivers.SAFE.getOSV()*: new argument *returnMatch* to also return the name of an OSV file instead of just downloading it.

SNAP API

- arbitrary nodes can now be parsed. Before, only a small selection of nodes (those used by function *geocode()*) were available. Now, any node and its default parametrization can be parsed to XML from the GPT documentation by internally calling e.g.:

  ```
  gpt Terrain-Flattening -h
  ```

  The parsed XML representation is saved for faster future reuse. See function *parse_node()* for details. In all cases the standard SNAP file suffix is used for output products, e.g. *_TF* for *Terrain-Flattening*.
- multi-source nodes like *SliceAssembly* now take any number of sources, not just two. See class *Node*.
- function *pyroSAR.snap.util.geocode()*:
  - new argument *nodataValueAtSea* to decide whether sea areas are masked out. Depends on the quality of the sea mask in the input DEM.
  - automatically download required Sentinel-1 Orbit State Vector (OSV) files.
  - new argument *allow_RES_OSV* to decide whether to allow usage of the less accurate Sentinel-1 RES OSV files in case the POE file is not available yet.
  - new argument *demName* to choose the type of the auto-downloaded DEM.

Auxiliary Data Handling

- class *pyroSAR.S1.OSV*:
  - removed progressbar from method *catch()* and made it optional in method *retrieve()* with new argument *pbar*

general

- bug fixes, new automated tests, documentation improvements

3.2.9 0.11.1 | 2020-07-17

- bug fixes

GAMMA API

- *Command API* compatibility with GAMMA version 20200713

3.2.10 0.12 | 2021-02-19

Drivers

- *pyroSAR.drivers.Archive*:
  - new argument *cleanup* to automatically remove missing scenes from database on initialization
  - method *insert()*: improved insertion speed
  - method *select_duplicates()*: new argument *value*
  - method *get_colnames()*: new argument *table* to get column names from arbitrary tables, not just the main *data* table
  - method *drop_element()*: option to remove scene from *data* and *duplicates* tables simultaneously by removing argument *table* and adding argument *with_duplicates*
  - method *drop_table()*:
    - new argument *verbose*
    - remove arbitrary tables, not just *data* and *duplicates*
  - method `drop_database()`: replaced by new function *pyroSAR.drivers.drop_archive()*
  - new method *add_tables()* to add custom tables to a database
  - bug fixes
- *pyroSAR.drivers.CEOS_PSR*:
  - added support for ALOS-1 PALSAR
  - added basic support for Level 1.0 data
- *pyroSAR.drivers.SAFE*:
  - method *getOSV()*: new argument *useLocal* to not search online if local matching files are found

GAMMA API

- *Command API* compatibility with GAMMA version 20201216
- function *pyroSAR.gamma.convert2gamma()*:
  - renamed argument *S1_noiseremoval* to *S1_tnr* (thermal noise removal)
  - new argument *S1_bnr* (border noise removal)
- function *pyroSAR.gamma.geocode()*:
  - new default `removeS1BorderNoiseMethod='gamma'`
  - renamed argument *tempdir* to *tmpdir*

SNAP API

- function *pyroSAR.snap.util.geocode()*:
  - enable grid alignment with new arguments *alignToStandardGrid*, *standardGridOriginX* and *standardGridOriginY*
  - new argument *tmpdir* to choose the location of temporarily created files
  - bug fixes
- function *pyroSAR.snap.auxil.gpt()*:
  - perform custom pyroSAR S1 GRD border noise removal only if IPF<2.9

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_autoload()*: return *None* if a VRT was defined

3.2.11 0.12.1 | 2021-03-09

SNAP API

- function *pyroSAR.snap.util.geocode()*:
  - output both sigma0 and gamma0 via argument *refarea*
  - new *export_extra* option 'layoverShadowMask'
- numerous bug fixes and API improvements

Auxiliary Data Handling

- class *pyroSAR.S1.OSV*:
  - download files from https://scihub.copernicus.eu/gnss

3.2.12 0.13 | 2021-09-10

Drivers

- new class *pyroSAR.drivers.EORC_PSR*
- new argument *exist_ok* for ID object unpack methods to enable reuse of already unpacked scenes
- *pyroSAR.drivers.SAFE.getOSV()*: new argument *url_option* to choose between different download URLs
- *pyroSAR.drivers.SAFE* align coordinate sorting of attribute *meta['coordinates']* with CRS description
- *pyroSAR.drivers.identify_many()*: disable progressbar by default

GAMMA API

- adaptations to enable processing of *EORC_PSR* data:
  - *pyroSAR.gamma.calibrate()*
  - *pyroSAR.gamma.convert2gamma()*
  - *pyroSAR.gamma.geocode()*
- *pyroSAR.gamma.geocode()*:
  - experimental optional refinement of the geocoding lookup table with new argument *refine_lut*
  - removed arguments *normalization_method*, *func_interp*, *removeS1BorderNoise*, *sarSimCC*
  - limit radiometric normalization to RTC correction method
  - simplify and improve computation of RTC contribution area
  - file suffices *pan* and *norm* have been replaced with *gamma0-rtc*
  - argument *export_extra* options:
    - removed *pix_geo*
    - renamed *pix_fine* to *pix_ratio*
    - added *pix_area_sigma0*, *pix_area_sigma0_geo*, *pix_area_gamma0_geo*, *gs_ratio* , *gs_ratio_geo*, *pix_ratio_geo*
  - use a dedicated temporary directory to unpack the scene and write GAMMA files so that they are separated (the GAMMA files used to be written to the unpacked scene's directory)
  - enable multiple scenes as input so that they can be mosaiced in SAR geometry before geocoding
- *pyroSAR.gamma.correctOSV()*: new argument *directory*
- *pyroSAR.gamma.multilook()*: new argument *exist_ok*
- *pyroSAR.gamma.convert2gamma()*: new argument *exist_ok*
- function *pyroSAR.gamma.dem.dem_autocreate()*:
  - do not apply an extent buffer by default
  - allow geometry in arbitrary CRS

SNAP API

- function *pyroSAR.snap.util.geocode()*:
  - new *export_extra* option *scatteringArea*
- extended support for *BandMaths* operator

Auxiliary Data Handling

- method *pyroSAR.S1.OSV.catch()*: new argument *url_option* with two download URLs to choose from
- function *pyroSAR.auxdata.dem_autoload()*:
  - added new DEM option *GETASSE30*
  - align pixels of subsetted VRT with original tiles
- function *pyroSAR.auxdata.dem_create()*:
  - new argument *outputBounds*

general

- replaced print messages with logging. This made the *verbose* argument that was used by several functions and methods obsolete; affects the following:
  - *pyroSAR.drivers.identify_many()*: replaced by argument *pbar*
  - *pyroSAR.drivers.Archive.add_tables()*: removed
  - *pyroSAR.drivers.Archive.drop_table()*: removed
  - *pyroSAR.drivers.Archive.insert()*: replaced by argument *pbar*
  - *pyroSAR.drivers.Archive.import_outdated()*: removed
  - *pyroSAR.drivers.Archive.move()*: replaced by argument *pbar*
  - *pyroSAR.drivers.Archive.select()*: removed
  - *pyroSAR.snap.auxil.execute()*: removed

  See section *Logging* for details.

3.2.13 0.14.0 | 2021-10-12

Drivers

- raise more appropriate errors (c430c59)
- `pyroSAR.drivers.findfiles()`: removed (functionality contained in *pyroSAR.drivers.ID.findfiles()*, now making use of `spatialist.ancillary.finder()`)
- *pyroSAR.drivers.Archive.select()*:
  - show progressbar for scene identification if `pbar=True`
  - enabled input of `datetime` objects for arguments `mindate` and `maxdate`
- *pyroSAR.drivers.identify_many()*: issue a warning when a file cannot be accessed (instead of raising a `PermissionError`)

GAMMA API

- *pyroSAR.gamma.dem.dem_autocreate()*: support for new DEM options provided by *pyroSAR.auxdata.dem_autoload()*

SNAP API

- `pyroSAR.snap.auxil.get_egm96_lookup()` removed in favor of new function *pyroSAR.auxdata.get_egm_lookup()*

Auxiliary Data Handling

- method *pyroSAR.S1.OSV.retrieve()*: thread-safe writing of orbit files
- new function *pyroSAR.auxdata.get_egm_lookup()*
- function *pyroSAR.auxdata.dem_create()*
  - new geoid option 'EGM2008'
  - make use of *get_egm_lookup()* for auto-download of EGM lookup files
  - several bug fixes related to vertical CRS transformation

  - bug fix for target pixel alignment
- function *pyroSAR.auxdata.dem_autoload()*: new DEM options:
  - 'Copernicus 10m EEA DEM'
  - 'Copernicus 30m Global DEM'
  - 'Copernicus 90m Global DEM'

general

- replaced http URLs with https where applicable
- improved documentation

3.2.14 0.15.0 | 2022-01-04

Drivers

- *pyroSAR.drivers.ID.geometry()*: new method

GAMMA API

- *Command API* compatibility with GAMMA version 20211208
- renamed argument *resolution* to *spacing*; affects:
  - *pyroSAR.gamma.geocode()*
  - *pyroSAR.gamma.ovs()*
  - *pyroSAR.gamma.multilook()*
- function *pyroSAR.gamma.calibrate()*
  - removed argument *replace*
  - added argument *return_fnames*
- function *pyroSAR.gamma.convert2gamma()*
  - added argument *return_fnames*
- function *pyroSAR.gamma.multilook()*
  - pass multiple Sentinel-1 sub-swaths to argument *infile* which are then combined into a single MLI using GAMMA command *isp.multi_look_ScanSAR*
- class *pyroSAR.gamma.ISPPar*:
  - new object attribute *filetype* with possible values 'isp' and 'dem'

SNAP API

- function *pyroSAR.snap.util.geocode()*:
  - enabled SLC processing
  - enable processing of sigma nought RTC
  - new *export_extra* argument *gammaSigmaRatio*
  - simplified workflow by writing layover-shadow mask directly from *Terrain-Correction*
  - changed processing node sequence:

        * was: Read->ThermalNoiseRemoval->SliceAssembly->Remove-GRD-Border-Noise->Calibration
        * is: Read->Remove-GRD-Border-Noise->Calibration->ThermalNoiseRemoval->SliceAssembly
    - new output image naming scheme, e.g.
        * S1A__IW___A_20210914T191350_VV_gamma0-rtc.tif
        * S1A__IW___A_20210914T191350_VH_sigma0-elp.tif
- function *pyroSAR.snap.auxil.gpt()*:
    - removed argument *multisource*
    - added argument *tmpdir*

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_autoload()*:
    - updated version of 'Copernicus 10m EEA DEM' from '2020_1' to '2021_1'
    - new DEM options:
        * 'Copernicus 30m Global DEM II'
        * 'Copernicus 90m Global DEM II'

general

- compatibility with sqlalchemy>=1.4

3.2.15 0.15.1 | 2022-01-07

general

- bug fixes

3.2.16 0.16.0 | 2022-03-03

Drivers

- *pyroSAR.drivers.BEAM_DIMAP*: new driver supporting SNAP's BEAM-DIMAP format
- *pyroSAR.drivers.SAFE*:
    - corrected SLC metadata (was read from first sub-swath, now from center sub-swath or as sum of all sub-swaths): center: spacing, heading, incidence; sum: samples, lines
    - new property *pyroSAR.drivers.SAFE.resolution*

Auxiliary Data Handling

- create water body mask mosaics from ancillary DEM products. Affects the following:
  - function *pyroSAR.auxdata.dem_autoload()*: new arguments *nodata* and *hide_nodata*
- function *pyroSAR.auxdata.dem_create()*:
  - new arguments *pbar* and *threads*

SNAP API

- new method *pyroSAR.snap.auxil.Par_BandMath.add_equation()*
- new function *pyroSAR.snap.util.noise_power()*
- new function *pyroSAR.snap.auxil.erode_edges()*
- function *pyroSAR.snap.auxil.writer()*:
  - new arguments *clean_edges* and *clean_edges_npixels* (to make use of function *erode_edges()*)
  - enabled conversion of BEAM-DIMAP files
- function *pyroSAR.snap.util.geocode()*:
  - new arguments *clean_edges* and *clean_edges_npixels* (see function *writer()*)
  - renamed argument *tr* to *spacing*
  - new arguments *rlks* and *azlks* to manually set the number of looks

GAMMA API

- function *pyroSAR.gamma.geocode()*:
  - new arguments *rlks* and *azlks*
- function *pyroSAR.gamma.multilook()*:
  - new arguments *rlks* and *azlks*

general

- correction of multi-look factor computation. Before: approximate target pixel spacing but never exceed it. Now: first best approximate the azimuth spacing as close as possible (even if this means exceeding the target spacing) and then choose the range looks to approximate a square pixel as close as possible. API changes:
  - function *pyroSAR.ancillary.multilook_factors()*:
    - renamed argument *sp_rg* to *source_rg*
    - renamed argument *sp_az* to *source_az*
    - replaced arguments *tr_rg* and *tr_az* with unified *target*

3.2.17 0.16.1 | 2022-03-07

Auxiliary Data Handling

- function *pyroSAR.auxdata.get_egm_lookup()*:
  - changed URL for PROJ geoid models, which results in better performance for function *pyroSAR.auxdata.dem_create()* (See pyroSAR#200).

3.2.18 0.16.2 | 2022-03-14

SNAP API

- function *pyroSAR.snap.util.noise_power()*: added missing orbit state vector refinement

3.2.19 0.16.3 | 2022-03-23

SNAP API

- function *pyroSAR.snap.util.noise_power()*: pass argument *cleanup* to *gpt()* call
- function *gpt()*: shortened names of temporary directories
- function *erode_edges()*: fixed bug in polygon selection
- function *writer()*: do not erode edges of layover-shadow mask

3.2.20 0.17.0 | 2022-05-30

SNAP API

- function `pyroSAR.snap.erode_edges()`: reuse mask for all images

GAMMA API

- new function *pyroSAR.gamma.dem.dem_import()*
- function *pyroSAR.gamma.geocode()*:
  - new argument *update_osv*

general

- full support for Sentinel-1 stripmap mode; renamed *SM* naming pattern to *S1..S6* to differentiate different beams
- bug fixes

3.2.21 0.17.2 | 2022-06-23

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_create()*:
  - use maximum possible value of *dtype* (e.g. 255 for unit8) instead of -32767.0 if the nodata value cannot be read from the source file
  - always use the same value for source and destination nodata

3.2.22 0.17.3 | 2022-07-03

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_create()*:
  - In case the nodata value could not be read from the source file, the function used to define a value itself, which is prone to errors. This value now needs to be set by a user via new argument *nodata* if it cannot be read from the source file.
  - bug fix: no longer try to download 'Copernicus 30m Global DEM' or 'Copernicus 90m Global DEM' tiles that don't exist.
- function *pyroSAR.auxdata.dem_autoload()*:
  - new argument *dst_nodata*. This can be used to temporarily override the native nodata value for extrapolation of ocean areas (in combination with `hide_nodata=True`).

3.2.23 0.18.0 | 2022-08-24

Drivers

- method *pyroSAR.drivers.SAFE.quicklook()*: new argument *na_transparent*
- new class *TDM*
- method `pyroSAR.drivers.TSX.getCorners()`: fixed bug in longitude computation
- class *ESA*: improved support for ERS and ASAR

GAMMA API

- *Command API* compatibility with GAMMA version 20220629

SNAP API

- compatibility with SNAP version 9
- function *geocode()*: improved support for ERS and ASAR

3.2.24 0.19.0 | 2022-09-28

Drivers

- class *pyroSAR.drivers.ESA*: added support for ASAR WSM

SNAP API

- new convenience functions:
  - *pyroSAR.snap.auxil.geo_parametrize()*
  - *pyroSAR.snap.auxil.sub_parametrize()*
  - *pyroSAR.snap.auxil.mli_parametrize()*
  - *pyroSAR.snap.auxil.dem_parametrize()*
- function *pyroSAR.snap.auxil.orb_parametrize()*: removed args *workflow*, *before*, *continueOnFail*; added *kwargs*
- function *pyroSAR.snap.auxil.erode_edges()*: extended to also take a BEAM-DIMAP product as input or a folder of multiple ENVI files (and not just and individual ENVI file)
- function *pyroSAR.snap.auxil.Workflow.insert_node()*: option to insert multiple nodes at once

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_autoload()*:
  - new argument *crop* to optionally return the full extent of all overlapping DEM tiles
  - added download status print messages
  - download and modify a Copernicus DEM index file for future reuse; this removes the need to search the FTP server for files and thus greatly accelerates the process of collecting all files overlapping with the AOI

3.2.25 0.20.0 | 2022-12-27

Drivers

- class *pyroSAR.drivers.ESA*: changed ASAR orbit type from DELFT to DORIS
- class *pyroSAR.drivers.BEAM_DIMAP*: new attributes *meta['incidence']* and *meta['image_geometry']*
- class *pyroSAR.drivers.Archive*: new argument *date_strict* for method *select()*

SNAP API

- function *pyroSAR.snap.util.geocode()*: force multi-looking for ERS1, ERS2, ASAR even if range and azimuth factor are both 1

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_autoload()*:
  - no longer require DEM tiles for creating a mosaic to address ocean cases
  - simplified handling and removed arguments *nodata*, *dst_nodata* and *hide_nodata*
  - the DEM option 'Copernicus 30m Global DEM' now also includes several auxiliary layers that can be downloaded automatically
  - the URLs for DEM options 'SRTM 3Sec' and 'TDX90m' have been updated
- function *pyroSAR.auxdata.dem_create()*:
  - option to customize the output DEM via additional keyword arguments to be passed to `spatialist.auxil.gdalwarp()`
  - no longer require a nodata value

3.2.26 0.21.0 | 2023-05-11

Drivers

- class *pyroSAR.drivers.Archive*:
  - improved PostgreSQL connection stability
  - method *select()*: the *vectorobject* geometry is now cloned before being reprojected to EPSG:4326 so that the source geometry remains unaltered

GAMMA API

- the *LAT* module is no longer needed: new pyroSAR-internal implementations can be used if the module is missing (concerns commands *product*, *ratio* and *linear_to_dB*)
- improved backwards compatibility:
  - use *multi_look_ScanSAR* if present and *multi_S1_TOPS* otherwise
  - use *gc_map2* if possible (present and with all needed arguments) and *gc_map* otherwise
  - addressed the case where *gc_map* does not have an argument *OFF_par*
- function *gamma.pixel_area_wrap*: new argument *exist_ok* (this function will be made more visible in the documentation once matured)
- bug fixes:
  - *pyroSAR.gamma.convert2gamma()*: raise an error if *S1_bnr=True* but the GAMMA command does not support border noise removal
  - *pyroSAR.gamma.geocode()*: removed unneeded underscore in HDR file naming
  - *gamma.pixel_area_wrap*: fixed some issues with occasionally missing intermediate files, e.g. for computing ratios

SNAP API

- function *pyroSAR.snap.util.geocode()*: new argument *dem_oversampling_multiple* with default 2 to increase the DEM oversampling factor for terrain flattening
- function *pyroSAR.snap.auxil.erode_edges()*:
  - do not attempt to perform erosion if the image only contains nodata (this might happen if only parts of the image were geocoded)
  - make sure that a backscatter image is used for erosion (auxiliary data like the local incidence angle often has a larger valid data extent and using such image for erosion would thus not properly erode edges of the backscatter images; additionally this has the effect that all images will have the same valid data extent after erosion)
  - the written mask files (delineating valid data and nodata after erosion of the backscatter image and used for masking all other images) are now compressed (deflate) so that data volume is decreased significantly

Auxiliary Data Handling

- function *pyroSAR.auxdata.dem_create()*:
  - new argument *resampleAlg* to change the resampling algorithm

3.2.27 0.22.0 | 2023-09-21

Drivers

- class *pyroSAR.drivers.Archive*:
  - allow multiple products with same *outname_base*, e.g. Sentinel-1 GRD and SLC; this required the introduction of a second primary key in the database
  - method *import_outdated()*: option to import data from an old database with only one primary key; this requires the old database to be opened in legacy mode (new argument *legacy=True*)
- class *pyroSAR.drivers.SAFE*: support for handling Sentinel-1 OCN products (metadata reading and database handling)

Auxiliary Data Handling

- class *pyroSAR.auxdata.DEMHandler*: enabled handling of southern hemisphere geometries.

3.2.28 0.22.1 | 2023-10-11

Drivers

- class *pyroSAR.drivers.BEAM_DIMAP*: enable calling inherited method *geometry()*

3.2.29 0.22.2 | 2023-11-16

SNAP API

- function *pyroSAR.snap.auxil.writer()*: fixed bug in ignoring *erode_edges* argument
- function *pyroSAR.snap.auxil.erode_edges()*: enable handling of polarimetric matrices

Drivers

- function *pyroSAR.drivers.identify()*: enable reading of *TDM* products

Misc

- class *pyroSAR.examine.ExamineGamma*: enhanced flexibility in finding GAMMA installation

3.2.30 0.23.0 | 2023-11-23

Drivers

- class *pyroSAR.drivers.Archive*: fixed bug in loading spatialite on Darwin-based systems

Auxiliary Data Handling

changes to Sentinel-1 OSV data handling:

- method *pyroSAR.S1.OSV.catch()*:
  - removed *url_option* 1 (https://scihub.copernicus.eu/gnss)
  - made option 2 the new default option 1 (https://step.esa.int/auxdata/orbits/Sentinel-1)
- added new arguments to the following functions:
  - *pyroSAR.gamma.correctOSV()*: *url_option*
  - *pyroSAR.gamma.geocode()*: *s1_osv_url_option*
  - *pyroSAR.snap.auxil.orb_parametrize()*: *url_option*
  - *pyroSAR.snap.util.geocode()*: *s1_osv_url_option*
  - *pyroSAR.snap.util.noise_power()*: *osv_url_option*

3.2.31 0.24.0 | 2024-01-10

Drivers

- new base attribute *coordinates*
- enable method *geometry()* for all driver classes
- classes *ESA* and *CEOS_ERS*: removed call to *gdalinfo* (for increased test capability and speed)
- outsourced regular expressions for product identification into separate module *patterns*

Auxiliary Data Handling

- method *pyroSAR.S1.OSV.catch()*: fixed bug in finding files starting in previous month

3.2.32 0.25.0 | 2024-04-16

Drivers

- class *pyroSAR.drivers.Archive*:
  - replaced column *bbox* with *geometry*; requires database migration
  - method *export2shp()*: improved column name laundering

SNAP API

- function *pyroSAR.snap.auxil.gpt()*: fixed bug that occurred during removal of BNR node

Ancillary Tools

- new classes *pyroSAR.ancillary.Lock* and *pyroSAR.ancillary.LockCollection* for custom file/folder locking

Auxiliary Data Handling

changes to Sentinel-1 OSV data handling:

- function *pyroSAR.auxdata.dem_create()*:
  - make use of new classes *Lock* and *LockCollection* for DEM download and mosaic creation (new argument *lock_timeout*)
  - check whether all VRT source files exist

3.3 Publications

- J. Truckenbrodt, F. Cremer, I. Baris, and J. Eberle. Pyrosar: a framework for large-scale sar satellite data processing. In P. Soille, S. Loekken, and S. Albani, editors, *Big Data from Space*, 197–200. Luxembourg, 2019. Publications Office of the European Union. doi:10.2760/848593.
- J. Truckenbrodt, T. Freemantle, C. Williams, T. Jones, D. Small, C. Dubois, C. Thiel, C. Rossi, A. Syriou, and G. Giuliani. Towards sentinel-1 sar analysis-ready data: a best practices assessment on preparing backscatter data for the cube. *Data*, 2019. doi:10.3390/data4030093.

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

[1] I. Ali, S. Cao, V. Naeimi, C. Paulik, and W. Wagner. Methods to remove the border noise from sentinel-1 synthetic aperture radar data: implications and importance for time-series analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3):777–786, 2018. doi:10.1109/Jstars.2017.2787650.

[2] N. Miranda and G. Hajduch. Masking "no-value" pixels on grd products generated by the sentinel-1 esa ipf. Report, CLS, 29 January 2018. URL: https://sentinel.esa.int/documents/247904/2142675/Sentinel-1-masking-no-value-pixels-grd-products-note.

[3] D. Small. Flattening gamma: radiometric terrain correction for sar imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 49(8):3081–3093, 2011. doi:10.1109/Tgrs.2011.2120616.

[4] M. Visvalingam and J. D. Whyatt. Line generalization by repeated elimination of points. *Cartographic Journal*, 30(1):46–51, 1993. doi:10.1179/caj.1993.30.1.46.

PYTHON MODULE INDEX

p

INDEX

Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W