lazInSAR
The InSAR software for the lazy ones

Dominique Derauw

October 2, 2025

1 Objective of the document

This very small and basic document is meant to introduce lazInSAR; a command line InSAR processor performing a classical InSAR processing from data reading to geoprojection in a single command line.

2 lazInSAR installation and description

lazInSAR is a fully automated InSAR processor included in the the ***AMSTerEngine*** . This latter one is a set of command line tools allowing to perform step-by-step InSAR processing. The ***AMSTerEngine*** is the core of the so called ***AMSTer toolbox*** that allows among others managing large SAR time series for displacement monitoring through MSBAS processing.

AMSTer toolbox can be fully installed and configured using the provided installer. If downloading the ***AMSTerEngine*** only, this one can be compiled separately after decompressing the archive launching the command `make` (or `make USEOPENMP=YES` if willing to activate parallel processing) from the `./InSAR/sources` directory of the decompressed archive. All the executables including lazInSAR will be found within the `./InSAR/bin` sub-directory.
Before compiling, some libraries need to be installed. This can be done using the following commands:

```
➾ sudo apt install libhdf5-dev
➾ sudo apt install libtiff-dev
➾ sudo apt install libgeotiff-dev
➾ sudo apt install libxml2-dev
➾ sudo apt install libfftw3-dev
➾ sudo apt install libgsl-dev
➾ sudo apt install libnetcdf-dev
➾ sudo apt install libomp-dev
```

⊶ sudo apt install snaphu

LazInSAR functioning is quite basic:
If launched with no parameters or with the `-h` option a brief help is displayed:

```
lazInSAR -h
Usage:
-1- lazInSAR masterImage slaveImage workingDir [DEMFilePath] [kmlPathFile] [options]
-2- lazInSAR master&SlaveImagesDir workingDir  [DEMFilePath] [kmlPathFile] [options]
-3- lazInSAR pathTo/InSARParameters.txt [options]
-4- lazInSAR pathTo/SBInSARParameters.txt [options]

For CSK and K5, masterImage and slaveImage are full path to corresponding .h5 files
For TSX, TDX, RS2 and SAOCOM, masterImage and slaveImage are full path to
directories containing the master and slave data.
For TDM, master&SlaveImagesDir is the full path to the TDM directory structure
master and slave images can also be given directly in previously read .csl image
format

For S1, either give path to uncompressed master and slave images, or give the path
to a directory containing any number of uncompresses S1 images.
In the latter case, all images will be read and converted in .csl format.
S1 read images can then be used in a second run, selecting directly master and
slave images in .csl format.

Some options are:
-topo : By default, lazInSAR is working in defo mode. So...
-coh : Skip filtering and unwraping. Aims at getting geoProjected coherence for
rapid proofing.
-skipUnwrapping: As its names states it
-skipFilter: As its names states it
-skipGeoProj: As its names states it
-SBInSAR: Split Band InSAR processing is performed
-initInSAR: Only InSAR initialization is performed. This allows modifying default
parameters at will
-iono: Performs a split band processing of both master and slave in low and high
sub-bands of one third of available bandwidth for further ionospheric decomposition
```

In fact there is not very much to say. Except perhaps how to handle the external DEM and how to help lazInSAR handling precise orbits in the case of Sentinel data.

2.1 Environment variable settings that can be of some help

Your `.bashrc` file can be modified, adding the following lines :

```
# SAR specific
LAZINSARPATH=${HOME}/WhereTheHell/DidIPut/ThisFuckingSoft
PATH="$PATH:${LAZINSARPATH}/bin"
export ENVISAT_PRECISES_ORBITS_DIR="${LAZINSARPATH}/DATA/Orbits/EnviSAT/DORIS"
export S1_ORBITS_DIR="$\{LAZINSARPATH}/DATA/Orbits/S1"
export EXTERNAL_DEM_DIR="${LAZINSARPATH}/DATA/DEMS"
export EARTH_GRAVITATIONAL_MODELS_DIR="${LAZINSARPATH}/DATA/EGM"
```

For sure, indicated directories must exist for the environment variables be effective. With this configuration, your saved external DEM will be found by their names and, if present at indicated locations, precise orbits will automatically be handled. Put lazInSAR binary in the corresponding `${LAZINSARPATH}/bin` directory to include it to your `${PATH}` environment variable.

2.2 External DEM

lazInSAR recognize two DEM format. A simple home-made one and ENVI formatted ones. Both must be ellipsoidal heights in Longitude-Latitude on WGS84.
The home-made one consist in aggregated SRTM tiles ordered in the mathematical order accompanied by a `.txt` file describing it. A small routine named `getSRTMDEM` is available to that purpose. If the `EXTERNAL_DEM_DIR` environment variable is set, your aggregated DEM with its description `.txt` file will be saved at the right location.
Remind you that lazInSAR is working considering ellipsoidal heights, not geoidal ones. Consequently, if available at path
`${EARTH_GRAVITATIONAL_MODELS_DIR}/EGM96/WW15MGH.DAC`,
`agregateSRTMTiles` will consider the EGM96 ellipsoidal heights to generate an ellipsoidal reference DEM.
To that purpose, the EGM96 file named `WW15MGH.DAC` can be donloaded at:
`https://web.archive.org/web/20130314064801/http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm96/binary/WW15MGH.DAC`
ENVI formatted DEMs must give ellipsoidal heights, be coded in float or in short integers and saved with their .hdr descriptor file within the external DEM directory. Normally it should work. However, cross the fingers just in case.
If your `EXTERNAL_DEM_DIR` environment variable is correctly set and pointing to the directory where the DEM's are saved, giving the name of the DEM as command line parameter is sufficient. No need to give the full access path.
Any way, if no DEM is given or if the given one is not found or not recognized, geoprojection process will be performed on WGS84 or on an inflated ellipsoïd up to the scene average height, if this one is given in the image headers.

2.3 Use of a .kml polygon

A kml polygon can be given as command line parameter. This kml must be a`.kml` file, not a `.kmz` file. It can be generated simply using Google Earth. Preferably, use simple polygons.
The kml will be used to limit the InSAR processing. The `.kml` location will be transformed in slant range azimuth and the circumscribed rectangle will be considered to restrict the InSAR processing. The same `.kml` will be used to constrain the geoprojection to the so defined area.
In case of Sentinel1 data, it is also used to restrict reading to the required burst only. In the case your area of interest delimited by your kml is at the crossing of two frames, just put your master frames in a master directory and your slave frames in a

slave directory. Send both directory to lazInSAR as master and slave location. Only requested burst in both frames will be read, performing an automatic stitching.

2.4 Orbit management

LazInSAR is currently managing Sentinel1 precise orbits. If these orbits are locally available within the directory defined in the environment variable, they will be handled at reading time and used properly in subsequent InSAR processing.
A routine named `updateS1Orbits` is provided to update the local data base, downloading orbits from the ESA COPERNICUS web site. This routine doesn't need any parameters.
Pay attention that the first use of `updateS1Orbits` may take some time as it will require downloading all the available `PRECISE` orbits as also all the `RESTITUTED` obits covering the period after the last available `PRECISE` orbit. It is then your responsibility to run `updateS1Orbits` from time to time to keep your Sentinel1 precise orbit local data base up to date.

2.5 Predefined values

InSAR parameters are predefined with respect to slant range azimuth resolution ratio to generate InSAR products with approximately square pixels.
Geoprojection will be performed on a UTM grid of 50x50m. Results are generated in ENVI format to allow opening them with QGIS.
After a few fully automatic runs, you should understand the processing logic and data organization.
If willing to run `lazInSAR` with your own processing parameters, run it first with the `-init` keyword. Change your parameters at will within the `InSARParameters.txt` and/or the `geoProjectionParameters.txt` file you'll find in somewhere in your working directory. Then, rerun `lazInSAR` giving the `InSARParameters.txt` as first parameter and using the option you may find useful.

Feedback most wanted at dderauw@sareos.be