InSAR/DInSAR User Manuel

UG.CSL.SAO.13001

Prepared by Dominique Derauw

Verified by

Released by

Document Change Record

Issue	Date	Comments

Distribution List

Not restricted

Table of content

Applicable documents

Reference documents

[1] De Rauw D., “Phasimétrie par Radar à Synthèse d’Ouverture : théorie et applications”, Ph.D. thesis, University of Liège, January 1999.

[2] De Rauw D., “Developements of an InSAR Processor”, Proc SPIE, 2584, pp 376-383, 1995.

[3] De Rauw D., “Phase unwrapping using coherence measurements”, Proc SPIE, 2584, pp 319-324, 1995.

[4] De Rauw D., “Coregistration and complex interpolation”, Proc. SPIE, 2584, pp 319-324, 1995

[5] De Rauw D. & Moxhet J., “Multiple image SAR interferometry”, Proc. FRINGE 96, ESA SP-406, pp 167-178, 1996.

[6] De Rauw D. & Barbier Chr., “Quality assessment of InSAR topographic mapping: The exemple of Belgium”, Proc. SPIE, 3497, 1998.

[7] De Rauw D. & Orban A., “Baseline combination for InSAR DEM atimetric resolution enhancement”, FRINGE symposium 2003,

http://earth.esa.int/fringe03/proceedings/papers/31_derauw.pdf

[8] Volkov Vv. & Zhu Y., “ Deterministic phase unwrapping in the presence of noise”, Optics Letters, Vol. 28, 22, pp 2156-2158, 2003.

I. Scope of the document

This document is a basic User Guide of the CSL Interferometric SAR processor named hereafter CSL InSAR Suite (CIS). It describes fundamental functionalities to perform a classical InSAR/DInSAR processing.

Calibration process of CosmoSkymed data is described.

Additionally, the use of an external DEM is explained. This one can be used either in the calibration process of CSK data or within an InSAR/DInSAR process.

II. CSL InSAR Suite

II.1 Introduction

This document is a short user manual explaining basic functions of the CSL InSAR Suite (CIS).

It is considered that the user is already familiar with SAR interferometry terminology and processing sequence.

CIS is a command line processor using flat text files as interfaces. In order to illustrate the software capabilities, the well-known Landers Earthquake DInSAR displacement measurement is reproduced in a step-by-step procedure.

II.2 Software installation

The software is delivered as a compressed archive containing this document and either the compiled routines being part of the suite (binaries for MacOSX or Linux UBUNTU) or the C sources.

II.2.1 C Sources distribution

The InSAR suite in itself is delivered as a compressed file named CSLInSARSuite.tar.gz.

Once uncompressed, it will have the following directory structure:

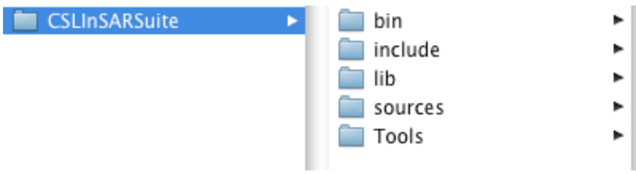


Figure 1: CSLInSARSuite directory structure

The bin directory is empty and will contain all routines (binaries) after compilation.

The include directory contains all the proprietary C headers.

The lib directory contains all the required proprietary C sources libraries.

The source directory contains main C sources for each routine.

Additionally, the source directory contains a basic makefile to perform compilation.

The Tools directory contains some convenient tools to be compiled independently. Each tool has its own subdirectory containing its own makefile.

To compile the suite, open a terminal and change directory to the source directory. Then type "`make`". Routines will be compiled and executables will be stored in the ./bin directory.

```
sources — bash — 72×39
62:~ dd$ cd /SAR/InSAR/sources
62:sources dd$ make

++ +++++++++++++++ ++
++ CSL InSAR suite ++
++ +++++++++++++++ ++
Library compilation ...

-- ------------------------ --
-- Data readers compilation --
-- ------------------------ --
-1- ERSDataReader
-2- EnviSATDataReader
-3- ALOSDataReader
-4- CSKDataReader
-5- TSXDataReader

-- ----------------------- --
-- InSAR suite compilation --
-- ----------------------- --
-1- initInSAR
-2- amplitudeImagesReduction
-3- coarseCoregistration
-4- fineCoregistration
-5- interpolation
-6- InSARProductsGeneration
-7- PolInSARProductsGeneration
-8- interferogramFiltering
-9- biasedCoherenceEstimation
-10- residuesSearch
-11- residuesConnexions
-12- phaseUnwrapping
-13- Deterministic phase unwrapping: detPhUn
-14- geoProjection
-15- DInSAR: removeTopographicPhase
-16- External DEM projection in Slant Range coordinates: slantRangeDEM
-17- CSK data calibration: SLC2MDG
62:sources dd$
```

Figure 2: CSLInSARSuite compilation

Routines are compiled in the order they should be used to perform a complete classical InSAR/PolInSAR processing.

After compilation, it is the responsibility of the user to store the binaries in a more convenient location and/or to change its PATH environment variable accordingly.

II.2.2 Remarks:

-1- In order to be able to handle CosmoSkymed data, the CSL InSAR Suite is provided with HDF5 libraries for compilation of the `CSKDataReader` routine under LINUX systems. If willing to compile on other systems, it is the user's responsibility to replace the provided libraries with those specifically compiled for the system under concern.

Locations of these libraries are shown on figure 3. If using a 32bit system (i386), corresponding libraries must be saved within the "32" sub-directory. If using a 64bit system (X86_64), corresponding libraries must be saved within the "64" sub-directory.

HDF5 sources and compilation instruction can be found on the HDF5 Group web site: http://www.hdfgroup.org/HDF5/Tutor/compile.html

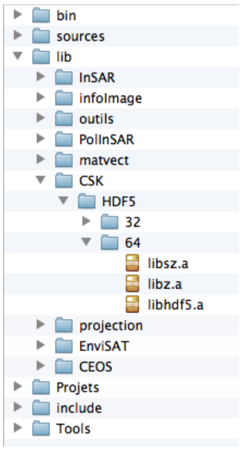


Figure 3: HDF5 libraries location

-2- In order to be able to handle TerraSAR-X data, XML libraries must be installed on the host system to allow compilation of the `TSXDataReader` routine.

To install this library under LINUX, just type the following command in a shell tool:

```
Sudo apt-get install libxml2-dev
```

II.2.3 Binaries distribution

Routines are either compiled for MacOSX 10.7/10.8 or for Linux Ubuntu systems.

Once the archive is uncompressed, it is the responsibility to copy executables in a convenient location adapting the PATH environment variable accordingly.

II.3 Common working description

Each routine as a simple help that can be displayed using the “`help`” keyword:

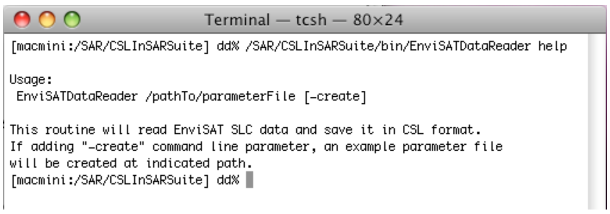


Figure 4: Help call

Each routine uses a parameters file as user interface.

For the `ERSDataReader`, `EnviSATDataReader`, `ALOSDataReader`, `CSKDataReader` , `TSXDataReader` and `initInSAR` routines, their respective parameters files may be created, launching the command with the path to the file to be created as first command line parameter along with the keyword “`-create`”

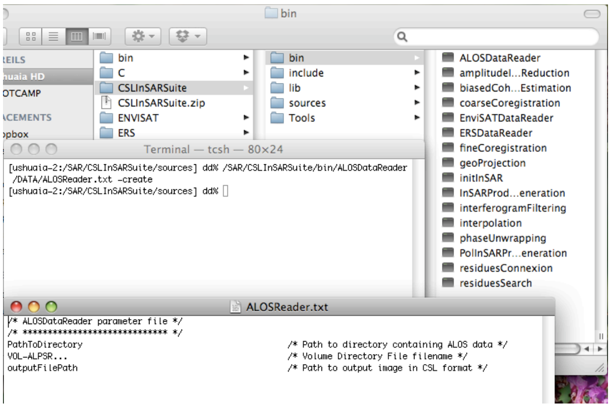


Figure 5: Example parameter file created with the ALOSDataReader routine

Whichever the parameter file, a parameter line is made of the parameter itself, starting at column 0 of the text file, followed by spaces and a comment.

The comment is opened by /* and closed by */, like for C comments.

The comment is used as keyword by the routine to find the corresponding parameters. Consequently, comments cannot be changed. However, the order of parameters lines within the parameter file is absolutely of no importance.

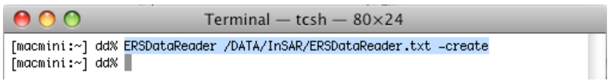
III. InSAR processing description

InSAR processing can be subdivided in three steps:

1. Data reading
2. InSAR initialization
3. InSAR processing

III.1 Data reading

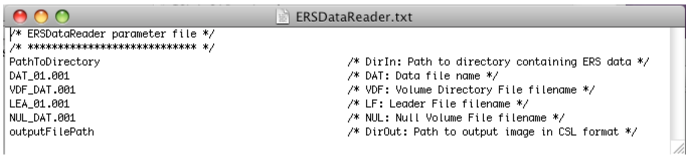
CIS handles ERS, EnviSAT, ALOS PalSAR, CosmoSkymed and TerraSAR-X (TSX & TDX) SLC data. Specific readers for each of these sensors are provided. Each routine require a specific parameters file. Path to the parameters file must be given as parameter to the reader routines. An example parameter file can be generated by the reader routines themselves using the "`-create`" keyword:



```
[macmini:~] dd% ERSDataReader /DATA/InSAR/ERSDataReader.txt -create
[macmini:~] dd%
```

Figure 6: Creation of an example parameter file for the ERSDataReader routine

The created text file must be opened and modified in order for the reader to find the input data and save it at the correct location.



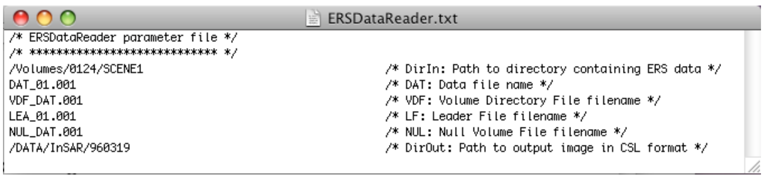
```
/* ERSDataReader parameter file */
/* *************************** */
PathToDirectory                 /* DirIn: Path to directory containing ERS data */
DAT_01.001                      /* DAT: Data file name */
VDF_DAT.001                     /* VDF: Volume Directory File filename */
LEA_01.001                      /* LF: Leader File filename */
NUL_DAT.001                     /* NUL: Null Volume File filename */
outputFilePath                  /* DirOut: Path to output image in CSL format */
```

Figure 7: Example parameter file created for the ERSDataReader routine

To read ERS data, the routine `ERSDataReader` needs the path to the ERS data directory which must contains the data itself and the headers, i.e. the Volume Directory File and the Leader File.

The Null Volume File header is not mandatory. If not found, the routine will simply prompt it and continue reading.

The output file path must be given, including the name the user wants to give to the read data:



```
/* ERSDataReader parameter file */
/* *************************** */
/Volumes/0124/SCENE1            /* DirIn: Path to directory containing ERS data */
DAT_01.001                      /* DAT: Data file name */
VDF_DAT.001                     /* VDF: Volume Directory File filename */
LEA_01.001                      /* LF: Leader File filename */
NUL_DAT.001                     /* NUL: Null Volume File filename */
/DATA/InSAR/960319              /* DirOut: Path to output image in CSL format */
```

Figure 8: Modified example parameter file for the ERSDataReader routine

Once the text file is modified, the routine can be launched using the same command than the one shown in figure 3, removing simply the "-create" keyword. The data will be read, converted and saved at the given location in CSL image format. Conversion consist in converting complex numbers coded in short integers for the real and imaginary parts (2 x 2 bytes per pixels) into complex numbers coded in real (`float`) numbers for the real and imaginary parts (2 x 4 bytes per pixels).

Remark: It must be noted that the data is saved using the format of the host machine. It is considered that the ERS or EnviSAT input data are given in Big Endian format. If the host machine on which runs the reader is an Intel-based (x86 - Low endian) architecture, bytes will be swapped.

In the case of ALOS, data are generally already delivered in float-float format. Therefore, conversion will only consist in swapping bytes if required by the host machine.

III.2 The CSL image Format

The CSL image format is a container of data, headers and text files that will be used as input for all further processing. The reader will add a “.csl” extension to the given output file name and create a directory of that name at the given location.

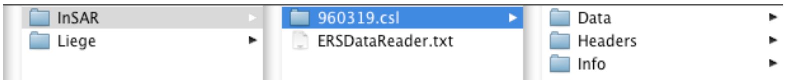


Figure 9:CSL image format structure

Within the CSL image container, there will be 3 subdirectories:

1. The Data directory contains the SLC data itself in raw format. It is thus simply the sequence of complex values for each pixels of the scene. Complex values are saved in (float;float) format. Data file name is always SLCData.XX where XX is the polarization configuration.
2. The Headers directory contains a copy of the input image headers. For ERS and ALOS, headers are simply copied into this directory. For EnviSAT, the header is extracted from the input data structure and saved in a single file. For CSK data, the headers are extracted for each HDF5 hierarchical data levels.
3. The Info directory contains a text file describing the SLC data contained in the Data directory and an information text file named SLCImageInfo.txt. This latter file gathers all parameters that will be required to perform the InSAR processing in itself. Additionally, it contains also the location of the scene on the reference ellipsoid found in the data header.

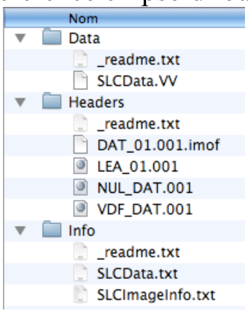


Figure 10: CSL image format structure and files

The CSL image format is the entry point of the CSL InSAR Suite.

Further readers will be developed to allow handling data from other sensors.

If several polarizations are available within the data, each channel is saved as a separated file within the Data subdirectory.

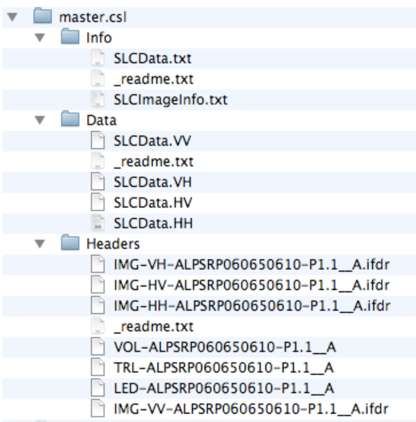


Figure 11: ALOS PALSAR Quad Pol data read and saved as “master” in CSL image format

III.3 InSAR initialization

Once both images of an interferometric pair have been read, InSAR initialization can start. The `initInSAR` routine requires a parameters text file whose location must be given as parameter. An example parameter file can be generated by the routine itself using the "`-create`" keyword.

The created parameters file must be modified to indicate the location of the master and slave images in CSL image format.

CIS is able to manage Bistatic interferometric pairs. Default parameter value is set to "`NO`". Turn it to "`YES`", if working with Tandem-X data.

The directory where the InSAR processing will take place must also be given. If this directory does not exist, it will be created.

initInSAR.txt

```
/* InSAR Initialization parameter file */
/* ********************************** */
/DATA/Vi-X/TDX/20110806.TDX.csl          /* Slave: Path to master image file in CSL format */
/DATA/Vi-X/TDX/20110806.TSX.csl          /* Master: Path to slave image file in CSL format */
YES                                      /* Bistatic interferometric pair [YES/NO] */
/DATA/Vi-X/TDX/i11Tandem0                /* Target directory: Path to directory for InSAR processing results */
```

Figure 12: InSAR initialization parameter file

The given target directory will contain three subdirectories:

1. The GeoProjection directory will contain the referencing files, the projection map and the geoprojected results.
2. The InSARProducts directory will contain all intermediate InSAR processing results up to the slant range DEM.
3. The TextFiles directory will contains four text files:
   a. The geoProjectionParameters.txt contains all the parameters required to perform geoprojection either in Longitude Latitude or in UTM.
   b. The InSARParameters.txt file contains all the required parameters to perform the InSAR processing.
   c. The two last files are copies of the SLCImageInfo.txt files of master and slave images.

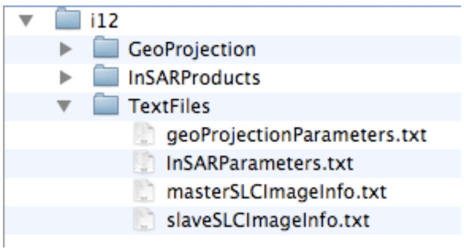


Figure 13: InSAR processing directory structure

Example InSAR and geoprojection parameters files are given in annex.

The InSARParameters.txt is the "user interface" to drive the processing. It is created with default parameters correctly suited to perform a classical InSAR processing of ERS or EnviSAT data set (beam 2). The user can modify all parameters. Examples of the `InSARParameters.txt` file and of the `geoProjectionParameters.txt` are given in Annex.

III.4 InSAR/PolInSAR Processing

After initialization, the InSAR parameters file will contain the computed baseline values as also an approximated transform to apply the slave image onto the master. Both the baseline and the transform are computed from state vectors given for each acquisition.

A classical InSAR/PolInSAR processing sequence is the following:

```
amplitudeImagesReduction
coarseCoregistration
fineCoregistration
interpolation
InSARProductsGeneration or PolInSARProductsGeneration
interferogramFiltering
biasedCoherenceEstimation
residuesSearch
residuesConnexions
phaseUnwrapping
geoProjection
```

After each process, the InSAR parameters file is updated to take into account the results of the performed processing step.

When a new file is issued, its dimensions are updated in the `InSARParameters.txt` file. Depending on the user system, the parameters file should be closed and reopened to see changes.

Concerning generated data, all issued files are in raw format. They contain no headers. Raw files are simply made of the sequence of pixel values. They can thus be open by any imaging software accepting raw images, providing their X and Y dimensions read in the updated `InSARParameters.txt` file.

Remark: ImageJ is a free software allowing to open easily generated images to view results, manipulate, measure and perform basic image processing.

It can be downloaded here: http://rsbweb.nih.gov/ij/download.html

III.4.1 amplitudeImagesReduction

This routine will compute the amplitude of both images and reduce the result by box averaging. Reduction factors are given in the parameters file and are set to 2x10 (range x azimuth pixels) by default.

Issued data files are made of real numbers coded in float (4 bytes per pixels) and saved within the ./InSARProducts subdirectory.

These images will be used for coarse coregistration process.

Remark:

`amplitudeImageReduction` can also be used using simply a single CSL image as first parameters. Reduction factors must then be given as second and third parameters. In this case the reduced amplitude image will be saved within the ./Data subdirectory of the CSL image structure. The name of the image will be SLCData.XX.mod. Description of the image (size and reduction factors) will be saved as a parameter file named modImageInfo.txt within the ./Info directory.

III.4.2 coarseCoregistration

This routine will perform the coarse coregistration of master and slave image and issue the corresponding transform. Coregistration is performed by correlation of amplitude windows centre on target anchor points regularly distributed on the master image.

Coarse coregistration parameters are the following:

```
/* -2- Coarse coregistration */
/* ************************ */
-2                                      /* Estimated range shift [pix] */
-380                                    /* Estimated azimuth shift [pix] */
64                                      /* Coarse coregistration range window size [pix] */
64                                      /* Coarse coregistration azimuth window size [pix] */
100                                     /* Coarse registration range distance between anchor points [pix] */
100                                     /* Coarse registration azimuth distance between anchor points [pix] */
0.500000000                             /* Coarse coregistration correlation threshold */
5.000000000                             /* Corelation peak range width threshold */
5.000000000                             /* Corelation peak azimut width threshold */
```

Figure 14: Coarse coregistration parameters

Estimated shifts between both images are computed during initialization and are given for indicative purpose only.

Correlation windows must have dimensions equal to a power of two to allow correlation using Fast Fourier Transform processing. Windows can be rectangular.

Correlation is characterized by a number between 0 (no correlation) and 1 (identical images). A threshold can be set to be more or less selective with respect to correlation results. Anchor points leading to a correlation below the threshold will not be taken into account for the transform calculation. Default value is 0.5.

Correlation is characterized by a peak centered on the coregistration shift. This peak has a width depending on the quality of the correlation. The better the correlation, the sharpest the peak. Once again, one can play with this parameter to select most well correlated anchor points. Default values are 5 x 5 pixel half width at half maximum.

If coregistration fails, it is often due to a bad estimate of the azimuth shift. A solution is to first enlarge the window size in that dimension.

The transform will be updated at the end of the process.

The process can be run several times to refine the coregistration. The updated transform will then be used as starting point to improve the coregistration.

Remarks:

-1- To help analyzing the coarse co-registration process, the user can add the keyword “-s” in the command line. In that case, results for each coregistered anchor points will be saved as an image sequence. It is to say that three files will be created within the InSARProducts subdirectory. These files are named masterWindow, slaveWindow and correlationWindow. Each window will be saved in sequence within these files.

They can be opened and visualize using for example the ImageJ free software. This allows exploring correlation pike characteristics with respect to cut-out master and slave windows.

-2- Co-registration process is performed correlating amplitude windows in the Fourier space. It is to say that windows are Fourier transformed and then multiplied. Result is then back Fourier transformed to analyse the correlation pike of both selected amplitude windows. By default, a cosine low-cut filter is applied in the Fourier space to lower background importance and perform a correlation mainly based on contouring of image features. The default filter width is 1/16 of the used window size. It is to say that if the chosen window size is 64 pixels width, the cosine low cut filter will be limited to the 4 first low frequencies.

If not willing to use this low cut filter, the user can simply add the keyword “-noLow” in the command line.

If willing to enlarge or reduce the filter, the user can use the keywords “low N”, where N is the fraction of the window leading to the filter width. In other words, “low 16” is the default and “low 1” will lead to a cosine filter on all the frequencies.

III.4.3 fineCoregistration

This routine will perform fine coregistration of images using maximization of local coherence as optimization criterion. Coregistration is thus performed using full resolution complex images.

Fine registration parameters are the following:

```
/* -3- Fine coregistration */
/* ********************** */
5                                   /* Fine coregistration range window size [pix] */
25                                  /* Fine coregistration azimuth window size [pix] */
3                                   /* Estimated range error [pix] */
3                                   /* Estimated azimuth error [pix] */
20                                  /* Fine registration range distance between anchor points [pix] */
20                                  /* Fine registration azimuth distance between anchor points [pix] */
0.500000000                         /* Fine coregistration correlation threshold */
```

Figure 15: Fine coregistration parameters

Window dimension centred on candidate anchor points must be odd values. Default dimensions are 5 x 25 (range x azimuth pixels)

The slave window is moved on the master window and coherence is computed for each shift in order to map the coherence around the anchor point position. The shift range is given by the "`Estimated range error`" and the "`Estimated azimuth error`" parameters.

Size of the mesh of candidate anchor points is given by the two next parameters.

A correlation threshold can also be set to only keep well-anchored points for the transform calculation.

The transform will be updated at the end of the process. The process can be run several times to refine the coregistration. The updated transform will then be used as starting point to improve the coregistration.

III.4.4 interpolation

After coregistration, the slave image must be interpolated with respect to the computed transform to make it superimposable to the master one and allow InSAR products computation.

No parameter is required. By default, the interpolated slave image will be saved within the ./InSARProducts directory. The user can change the filename and destination.

If the slave image is made of several polarization channels, the user can use the keyword `allChannels` to interpolate all channels using the same transform for each.

Remark:

A master transform can also be set within the `InSARParameters.txt` file. If it is the case, the master image will be interpolated using this transform and the slave image will be interpolated using composition of both transform. This is peculiarly useful when both images of an interferometric pair must be coregistered to a global master image.

III.4.5 InSARProductsGeneration

This routine will compute the amplitude images, the interferogram, a filtered interferogram and the coherence image at the same time. If adding the "`+std`" keyword in the command line, phase and height standard deviation estimates will also be computed. By default, all those products are computed on the whole intersection of master and slave images. If willing to limit the processing to a smaller area, simply change the "`Area of interest`" parameters.

If several polarization channels are available, the ones used for the current InSAR processing can be chosen changing the "`Master polarization channel`" and "`Slave polarization channel`" parameters within the InSAR parameter file.

```
InSARParameters.txt
/* InSAR parameters file */
/* ******************** */

/* Input images */
/* ************ */
/DATA/InSAR/ALOS/master.csl                          /* Master image file path [CSL image format] */
HH                                                   /* Master polarization channel */
1248                                                 /* Master image range dimension [pixels] */
18432                                                /* Master image azimuth dimension [rows] */
/DATA/InSAR/ALOS/slave.csl                           /* Slave image file path [CSL image format] */
VH                                                   /* Slave polarization channel */
1248                                                 /* Slave image range dimension [pixels] */
18432                                                /* Slave image azimuth dimension [rows] */
750333.250000                                        /* Master image minimum slant range */
750782.937500                                        /* Slave image minimum slant range */
9.368514                                             /* Slant range sampling [m] */
1269999744.000000                                    /* Radar carrier frequency [Hz] */

/* Area of interest */
/* **************** */
0                                                    /* Lower left azimuth coordinate */
64                                                   /* Lower left range coordinate */
18394                                                /* Upper right azimuth coordinate */
1247                                                 /* Upper right range coordinate */
```

Figure 16: Input image parameters and area of interest definition

InSAR products are computed using a moving window, then reduced using a box-averaging window. Reduction window size is given in the "`Box averaging`" section of the `InSARParameters.txt` file. Moving window size is given in the "`Interferometric products computation`" section through the coherence estimator window size.

```
/* -5- Interferometric products computation */
/* **************************************** */
0                                            /* Interferometric products range dimension [pix] */
0                                            /* Interferometric products azimuth dimension [pix] */

/DATA/InSAR/i12/InSARProducts/interfero          /* Interferogram file path */
/DATA/InSAR/i12/InSARProducts/interfero.f1x5     /* Filtered interferogram file path */
1.000000000                                  /* Range filter Full Width at Half Maximum */
5.000000000                                  /* Azimuth filter Full Width at Half Maximum */

/DATA/InSAR/i12/InSARProducts/coherence          /* Coherence file path */
2                                            /* Coherence estimator range window size [pix] */
10                                           /* Coherence estimator azimuth window size [pix] */
5                                            /* Square spiral size [pix] */
```

Figure 17: InSAR processing parameters

To improve coherence computation, an estimate of the topographic phase is computed and removed on each point before coherence estimation. This approximate topographic phase component is computed as the best local plane of phase. This latter one is computed after a local square spiral phase unwrapping. The user can modify the size of the square spiral. Default value is 5 pixels.

A filtered version of the interferogram is computed at the same time. Filter is applied to the full resolution interferogram before box averaging. Consequently, at this stage the filter size is given in full resolution pixels.

The filter is a Gaussian filter.

All issued InSAR products are made of real numbers coded in float (4 bytes per pixels).

Remark:

If an external DEM is made available, it will be automatically be taken into account and the routine will additionally issue a first phase component and a residual interferogram (see Chapter IV).

III.4.6 PolInSARProductsGeneration

PolInSAR processing is fully integrated within CIS. We consider that PolInSAR processing through coherence optimization is simply another way to get interferograms.

Consequently, if the user wants to make a PolInSAR processing, he has to follow the same preceding processing steps, considering one polarization channel as the leading one to coregister the whole slave data set and get the correct transform that will be applied to all channels. If the user considers that the channels are not necessarily aligned, he can perform the coregistration and interpolation process for each polarization channel independently.

Coherence optimization is performed on sub-windows whose dimensions are given through the "`Coherence estimator range window size`" and the "`Coherence estimator azimuth window size`" parameters.

To increase signal to noise ration, coherence optimization results are box averaged. The dimension of the box-averaging window are given through the "`Range reduction factor`" and "`Azimuth reduction factor`" parameters.

Coherence optimization will compute three optimized coherence images and associated interferograms.

Coherence optimization leads also to the computation of several side products: Amplitude images of each polarization channels are recomputed and HAα decomposition of each image is performed.

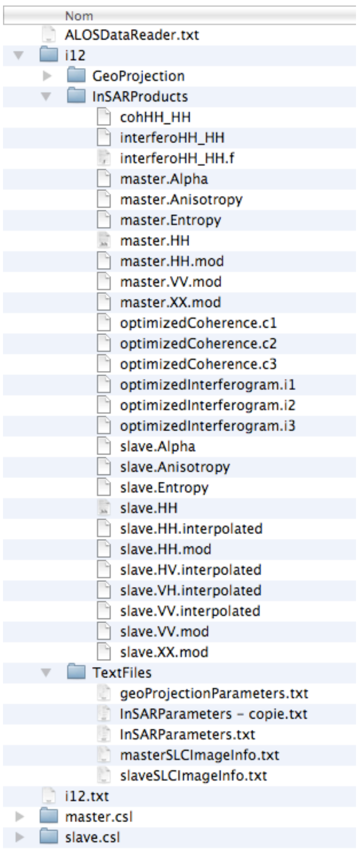


Figure 18: PolInSAR products listing

III.4.7 interferogramFiltering

If filtering issued by the former step appears to be inadequate or insufficient, an additional filtering may be applied.

Pay attention, this routine will also use the parameters given in the "`Interferometric products computation`" section. But in this case, it is the already computed interferogram that will be filtered and the result will be saved at the indicated path for the filtered interferogram. Since the computed interferogram is obtained through box averaging, the filter size must be adapted. Generally, a filter width of a fraction of a pixel gives already quite satisfactory results.

If it is the filtered interferogram that must be unwrapped, the user **must** change the interferogram file path by the one of the filtered one.

III.4.8 biasedCoherenceEstimation

This routine is the first to be used to perform the phase unwrapping. Implemented phase unwrapping is based on residues search and connexions (branch cuts algorithm).

Phase unwrapping process can be controlled setting parameters in the "`Phase unwrapping`" section of the `InSARParameters.txt` file.

The `biasedCoherence` routine computes a biased coherence that will be used to drive residues connexions.

Once again, an estimator window size and a square spiral size can be set.

Biased Coherence values are saved as real numbers coded in float (4 bytes per pixels)

```
/* -6- Phase unwrapping */
/* ******************* */
/DATA/InSAR/i12/InSARProducts/biasedCoherence         /* Biased coherence file path */
0                                                     /* Biased coherence range dimension [pix] */
0                                                     /* Biased coherence azimuth dimension [pix] */
5                                                     /* Biased coherence estimator range window size [pix] */
5                                                     /* Biased coherence estimator azimuth window size [pix] */
5                                                     /* Biased coherence square spiral size [pix] */

/DATA/InSAR/i12/InSARProducts/residus                 /* Residus image file path */
0                                                     /* Residus image range dimension [pix] */
0                                                     /* Residus image azimuth dimension [pix] */

/DATA/InSAR/i12/InSARProducts/connexions              /* Connexions image file path */
0                                                     /* Connexions image range dimension [pix] */
0                                                     /* Connexions image azimuth dimension [pix] */
2                                                     /* Range minimum search radius [pix] */
2                                                     /* Azimuth minimum search radius [pix] */
1                                                     /* Range search radius step [pix] */
1                                                     /* Azimuth search radius step [pix] */
1000                                                  /* Coherence scaling */
3                                                     /* Connexion process mode */
0.250000                                              /* Coherence cleaning threshold */
0.100000                                              /* False residue coherence threshold */
```

Figure 19: Phase unwrapping processing parameters

III.4.9 residuesSearch

This routine simply searches for residues within the interferogram.

Residues image is saved and coded in `unsigned char` (1 byte per pixels). Values are 0 for negative residues, 1 for no residue or 2 for positive residues.

III.4.10 residuesConnexions

This routine performs the connexions of residues to build branch cuts.

Residues connexion starts from the residue having the highest coherence. The connexion process is guided by the biased coherence image, trying to follow the lowest coherence path to connect residues belonging to the same phase discontinuity.

A connexion is made of the same number of positive and negative residues to be said equilibrated (having a null charge). Once a connexion is established, the next non-connected residue of highest coherence is chosen as new starting point.

A minimum search radius can be set. This allows searching first in the close surrounding of the starting residue to perhaps find a single residue of opposite value and build a short connexion made simply of a pair of residues.

If more than one residue is found in the minimal search ellipse, residue connexions continues looking for other residues up to completing a connexion containing as much negative than positive residues. The searched area is the intersection of an ellipse centred on each already found residue and the zone showing a biased coherence lower than a given threshold. The area of search is extended steps by steps increasing first the ellipse dimensions then increasing the biased coherence threshold.

Ellipse dimension increase can be controlled setting the "`Range search radius step`" and the "`Azimuth search radius step`" parameters.

Biased coherence threshold increase step can be controlled setting the "`Coherence scaling`" parameter. Biased coherence, which is a value between 0 and 1 is first scaled between 0 and this scaling parameters, then converted in `unsigned short` integer values. Default value is 1000, meaning that biased coherence threshold will be increased by steps of one thousandth.

Once a connexion has a null charge (the same number of negative and positive residues), the process can stop immediately or continue up to searching the whole low coherence area defined by the current coherence threshold. This behaviour is controlled by the "`Connexion process mode`" parameter. If set to 0, search will continue until the whole low coherence zone have been searched, increasing the search ellipse steps by steps. If sets to 1, search will immediately stops when an equilibrated connexion is established.

Any other integer value will indicate how many times the search ellipse must be increased before stopping. If after those steps, the connexion is still equilibrated (no other residue found), then the connexion process stops.

To avoid searching and performing very long connexions in areas of very low coherence, the "`Coherence cleaning threshold`" parameter can be set. All areas having a biased coherence below this threshold will not be searched provided they are connected to the border of the image. This is mainly useful when unwrapping a scene showing a coastline and some parts of open see.

The "`False residue coherence threshold`" parameter determines a biased coherence threshold under which all points will be considered as false residue having a null charge. So, they will not influence the charge of a connexion but they will help to better follow path of low coherence.

The connexion process issues two files: the connexion image and a superposition of the connexions and the biased coherence.

The connexion image is coded in `unsigned char` values. Connexions have a value of 255.

The superposition is coded in `unsigned short` values (2 bytes per pixels). The name of the superposition image file is the same than the one of the connexions image with the additional extension ".sup".

III.4.11 phaseUnwrapping

This routine will load the interferogram and the connexion images to perform the phase unwrapping by phase integration. Unwrapped phase will be saved and dimension parameters will be updated within the InSAR parameters file.

By default, the phase is unwrapped and converted to local heights. The slant range DEM is then saved and its dimensions are updated within the InSAR parameters file.

To only issue the unwrapped phase in radian, add the "`phaseOnly`" keyword. If using the "`crop`" keyword, InSAR products will all be cropped to the unwrapped phase image size.

III.4.12 detPhUn: Deterministic phase unwrapping

An alternative way to unwrap the phase is made available. It is a global phase unwrapping technique based on the frequency analysis of the image of phase gradients [8].

This routine can be used immediately after `InSARProductGeneration`. Residues, biased coherence and residue connection do not have to be searched or computed.

Once launched, this routine will perform a global phase unwrapping of the interferogram and create an unwrapped first phase component saved under this name by default (firstPhaseComponent) and having the same dimensions than the original interferogram. In addition, two other products are computed: a residual interferogram and an approximate slant range DEM.

The residual interferogram shows the remaining fringes that the algorithm did not succeed to unwrap. It generally shows a much lower fringe rate and can then be unwrapped classically using the residues method to improve the first phase component.

If a first phase component exists within the InSARProducts subdirectory, the classical phase unwrapping method will take it into account. It is to say that it is the residual interferogram that will be unwrapped and the issued unwrapped phase will be added to the first phase component to generate the unwrapped phase and the slant range DEM.

There is only one point the user must take care of, it is the filtered interferogram used to compute the biased coherence. At this stage, this latter one is well the classical interferogram filtered during InSAR product computation. If willing to compute conveniently the biased coherence linked to the residual interferogram, one must first filter it. To do so, put the name of the residual interferogram in place of the one of the full interferogram and filter it, giving a convenient name to its filtered version.

III.4.13 geoProjection

This routine performs the geoprojection of InSAR products using the geoProjectionParameters.txt file generated during the InSAR initialization (initInSAR).

This parameters file is supposed to be self explanatory for most parameters (see annex).

Two projections are available: Longitude-Latitude or UTM. By default the UTM mode is set in the parameter file.

Whatever the chosen projection, one can set an interpolation radius. This radius determines an ellipse around each target point on the destination grid into which intervening points will be searched. Intervening points are points in the initial geometry that will be taken into account to perform the extrapolation of the target point in the destination geometry.

```
/* UTM projection parameters */
/* ************************ */
100                         /* Easting sampling [m] */
100                         /* Northing sampling [m] */
70                          /* Easting interpolation radius [m] */
70                          /* Northing interpolation radius [m] */

/* Resampling parameters */
/* ******************** */
TRI                         /* Resampling method : TRI = Triangulation; AV = weighted average; NN = nearest neighbour */
ID                          /* Weighting method : ID = inverse distance; LORENTZ = lorentzian */
1.0                         /* ID smoothing factor */
1.0                         /* ID weighting exponent */
1.0                         /* FWHM : Lorentzian Full Width at Half Maximum */
/pathToMaskFile             /* Mask file */
-9999.                      /* Excluding height value */
-400.                       /* Minimal height value */
8000.                       /* Maximal height value */
```

Figure 20: Geo projection parameters

With respect to resampling process, several parameters can be set.

Three methods are provided: TRI, AV and NN.

TRI computes all possible triangles made of sets of three intervening points and containing the target point. From these triangulated values, a plane is computed and the value of the plane at the target position is considered as a good estimate to extrapolate the destination value. If several triangles are present, a weighted average of all these estimates is computed as the extrapolation of the target value. Used weight is the product of the weights of triangle summits with respect to target position. In this way, smallest triangles surrounding the target point the closest has the highest weight.

AV performs a simple weighted average.

NN considers simply the nearest neighbour as the interpolated value.

Two weighting are provided, ID and LORENTZ.

ID is the classical Inverse Distance weighting. This weighting function is given by:

$$weight = \frac{1}{(d+s)^e}$$

Where d is the distance between the intervening point and the target point, s is named the smoothing factor and e the exponent.

The LORENTZ weighting scheme uses a Lorentzian function given by:

$$weight = \frac{1}{1+\left(\frac{d}{\Gamma}\right)^2}$$

Where Γ is the Full Width at Half Maximum of the function.

One can control which DEM values are used in the extrapolation in three ways:

First, one can provide a mask image. This mask must have the same size than the slant range DEM used for the geoprojection. The mask must be coded in `unsigned char` values (1 byte per pixels). A value of 0 means that the point cannot be considered as an intervening point for the extrapolation. The coherence image may for example be used to generate the mask. No tools are provided to generate the mask.

A single excluding value can also be provided.

A third way to control the geoprojection is to provide an interval within which the heights values will be considered. Local phase errors may induce aberrant values in the unwrapped phase and consequently in the slant range DEM. This interval allows considering only height values that are a priori appropriate.

By default, the whole slant range DEM will be used to locate the geoprojection area. The whole slant range DEM will thus be projected. If willing to limit the geoprojection to a known area, the user can simply change the values of the considered frame in projected geometry units.

Remark: Longitude and latitude values must be given in decimal degrees. UTM Values are in meter.

IV. Use of an external DEM

IV.1 Generation of a slant range DEM

CIS is able to handle external DEMs given in longitude-latitude on WGS84.

The main routine to handle it is named: `slantRangeDEM`.

Like other routines, the keyword "-create" can be used to create an example parameter file:

```
/* Slant range DEM generation */
/* ************************** */
masterImagePath                        /* Reference slant range image path in CSL format */

/* External DEM characteristics */
/* **************************** */
File path                              /* Georeferenced DEM file path */
1000                                   /* X (longitude) dimension [pixels] */
1000                                   /* Y (latitude) dimension [pixels] */
5.5655                                 /* Lower left corner longitude [dd] */
50.5980                                /* Lower left corner latitude [dd] */
0.00027777777                          /* Longitude sampling [dd] */
0.00027777777                          /* Latitude sampling [dd] */

/* Resampling parameters */
0.                                     /* Range interpolation radius [pixel] */
0.                                     /* Azimuth interpolation radius [pixel] */
TRI                                    /* Resampling method : TRI = Triangulation; AV = weighted average; NN = nearest neighbour */
ID                                     /* Weighting method : ID = inverse distance; LORENTZ = lorentzian */
1.0                                    /* ID smoothing factor */
1.0                                    /* ID weighting exponent */
1.0                                    /* FWHM : Lorentzian Full Width at Half Maximum */

/* Reduction factors */
5                                      /* Range reduction factor */
5                                      /* Azimuth reduction factor */
```

Figure 21: slantRangeDEM example parameter file

The first parameter is the image (given in CSL format) for which the external DEM will be extracted to generate a corresponding slant range DEM.

Next section describes the external DEM given in input. External DEMs must be given in longitude-latitude. The software needs the size in pixels of the external DEM, the lower left coordinate and the sampling in both longitude and latitude (default value is 1" expressed in decimal degrees). Pay attention to the fact that for all CIS routines, all images and produced data are always expressed in a Cartesian coordinate system. It is the same for external data given in input. Consequently, the given external DEM must be saved from southern latitude to northern latitude.

Next section concerns the resampling method and parameters. Their meaning is the same than for the geoprojection process. Interpolation radii are set to zero by default and will be estimated from the slant range referencing process. The user can adapt these values to speed up the process or to try to fulfil some holes, extending these radii.

Last section gives the reducing factors. Default values are 5 pixels meaning that the slant range DEM will be computed every 5 pixels with respect to the full resolution SLC Data.

Once launched, the routine will extract from the external DEM, the area covered by the SLC image under concern. This part of the DEM will be saved under the name "externalDEM" within the ./Data subdirectory of the image in CSL format. An associated text file externalDEM.txt file will be saved within the ./Info subdirectory. This file contains simply the origin, the size and the sampling of the extracted DEM.

The extracted external DEM will be then projected in the slant range coordinate system of the chosen master image and saved under the name "slantRangeDEM" within the ./Data subdirectory of the image in CSL format. An associated text file slantRangeDEM.txt file will be saved within the ./Info subdirectory. This file is a copy of the parameter file with, in addition, the slant range projection frame and the size of the projected product.

An example obtained for a TerraSAR-X image of East Congo is shown in figure 22.

```
/* Slant range DEM generation */
/* ************************** */
/DATA/Vi-X/TDX/20110806.TSX.csl               /* Reference slant range image path in CSL format */

/* External DEM characteristics */
/* **************************** */
/DATA/Vi-X/SRTM_Rift                          /* Georeferenced DEM file path */
10801                                         /* X (longitude) dimension [pixels] */
21601                                         /* Y (latitude) dimension [pixels] */
27.99986111                                   /* Lower left corner longitude [dd] */
-5.99986111                                   /* Lower left corner latitude [dd] */
0.00027777777                                 /* Longitude sampling [dd] */
0.00027777777                                 /* Latitude sampling [dd] */

/* Resampling parameters */
24.969877                                     /* Range interpolation radius [pixel] */
28.924002                                     /* Azimuth interpolation radius [pixel] */
TRI                                           /* Resampling method : TRI = Triangulation; AV = weighted average; NN = nearest neighbour */
LORENTZ                                       /* Weighting method : ID = inverse distance; LORENTZ = lorentzian */
1.0                                           /* ID smoothing factor */
1.0                                           /* ID weighting exponent */
.5                                            /* FWHM : Lorentzian Full Width at Half Maximum */

/* Reduction factors */
4                                             /* Range reduction factor */
4                                             /* Azimuth reduction factor */


/* Slant range projection frame */
-20                                           /* xMin */
14068                                         /* xMax */
-20                                           /* yMin */
18176                                         /* yMax */

4                                             /* X sampling of projected product [pix] */
4                                             /* Y sampling of projected product [pix] */
3524                                          /* X size of projected products [pix] */
4551                                          /* Y size of projected products [pix] */

/* Additional affine transform */
/* x2 = Ax x1 + Bx y1 + Cx */
/* y2 = Ay x1 + By y1 + Cy */
1                                             /* Ax */
0                                             /* Bx */
0                                             /* Cx */
0                                             /* Ay */
1                                             /* By */
0                                             /* Cy */
```

Figure 22: slantRangeDEM.txt file

As can be seen in figure 22, an additional affine transform can be set and associated to the slant range DEM generated. The default transform is unitary.

Along with the slant range DEM, a simulated amplitude image of the area is generated on the sole basis of measured slopes on the external DEM projected in slant range.

Figure 23 shows an example of detected and simulated images for the TerraSAR-X acquisition of East Congo. The external DEM is a 1” SRTM DEM.

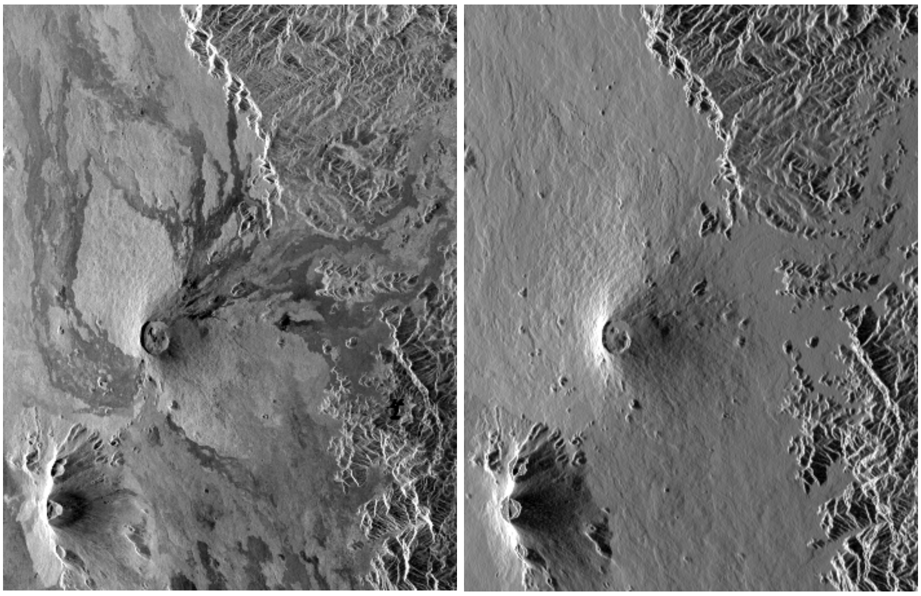


Figure 23: Detected (right) and simulated (left) image from an external DEM projected into slant range

The simulated image may be used to find the additional affine transform that may exists to link the external DEM with the image under concern. This affine transform may be computed using the coarse coregistration process, defining the real detected image as master and the simulated one as slave.

As can be seen in the figure here-above, orbitography given with the TerraSAR-X image is of high accuracy, leading immediately to a good external DEM projected in slant range. However, in the case of low precision orbits, like for ERS or ALOS images, post-registration of simulated and real image is mandatory to correctly link each slant range pixel to its correct height in the external DEM.

Remark:

Even in the case of the here above TerraSAR-X image, we found globally a slight translation of about 15 full resolution pixels in both dimensions (11 in range and 14.8 in azimuth). This shift corresponds to approximately 30m on ground; it is to say about 1”. Consequently, the registration error corresponds to one pixel of the original external DEM.

IV.2 InSAR/DInSAR processing using an external DEM

Once a master image contains a slant range DEM within its ./Data subdirectory with its corresponding description within its ./Info subdirectory, the `InSARProductGeneration` routine take it automatically into account. The slant range DEM will be used to generate a first phase component and a residual interferogram will be computed. The slant range DEM itself is resampled to fit with the generated InSAR products and a new slant range DEM is saved with the name given within the `InSARParameters.txt` file (see annexe).

If the used external DEM is of high accuracy, the residual interferogram can be considered as the differential one. If not, it can be unwrapped to be added to the first phase component in order to improve the external DEM.

The slant range DEM can also be used immediately to geoproject the InSAR products. To do so, simply run the `geoProjection` routine after generating the InSAR products.

Remark:

A very important remark must be done here. CIS always uses the exact mathematical expressions at all stages to compute a local height from an unwrapped phase or, conversely, a local phase from a known local height.

It has been observed that others softwares are using the (very good) first order approximation, computing an orbital phase and a local height of ambiguity to convert phase to height and vice versa.

Visibly, the SRTM DEM was computed using the same approximation. It is to say that to recover the correct topographic phase linked to the DEM, one must also use the same approximation. To do so, the `InSARProductGeneration` routine must be launched using the “-ha” keyword in the command line in order to force linear approximation of the topographic phase in the computation of the first phase component.

V. DInSAR processing step-by-step procedure

V.1 The Landers Earthquake case

A strong Earthquake took place on June 1992, near Landers, California. Thanks to ERS acquisitions, the JPL demonstrated the strong potential of differential SAR interferometry (DInSAR) to measure and map local displacement. They obtained their results using three images forming a coherent triplet. Out of this triplet, two SAR pairs were constituted. The first one made of two images acquired before (April 24, 1992) and after the Earthquake (July 8, 1992) were used to perform the displacement measurement. The second pair was used to generate a topographic phase to be removed from the first one.

Other teams, like CNES, reproduced their results using the same pair of images containing the displacement phase but using an external DEM to generate the topographic phase to be removed.

Like others, CSL reproduced these famous results to validate its DInSAR processor. In our case, we used an ERS Tandem pair acquired on January 7 and 8, 1996 to generate the topographic phase.

We will use this example processing to give a step-by-step procedure of both InSAR and DInSAR processing sequence.

V.2 Four images DInSAR

In four-images DInSAR, a SAR pair of images is considered as the displacement pair containing the movements we want to map and a second pair is considered as the topographic pair containing only a topographic phase. Each pair is made of a master and a slave image that must be coregistered to the master.

In addition, the two image pairs must be coregistered. Therefore, out of the four, one image is considered as the global master one.

In the following, we will consider the image of July 8, 1992 as being the global master.

Images will be numbered as follows:

1 : Acquisition of July 8, 1992

2 : Acquisition of April 24, 1992

3 : Acquisition of January 7, 1996

4 : Acquisition of January 8, 1996

With respect to this numbering, pair 12 is the displacement pair and pair 34 is the topographic pair. Pair 13 allows getting the coregistration transform between both pairs.

V.3 Step 0: Organize your work

The very first step consists in organizing your work. Create a directory hierarchy where you will save the input data and the InSAR results. In our case, we created a directory named Landers at the following location: `/Home/dd/DATA/InSARDemo/Landers`

V.4 Step 1: Data reading

First step consist in reading the data delivered in CEOS format. In our case, ERS data is available on CD and has the following structure (fig. 18):

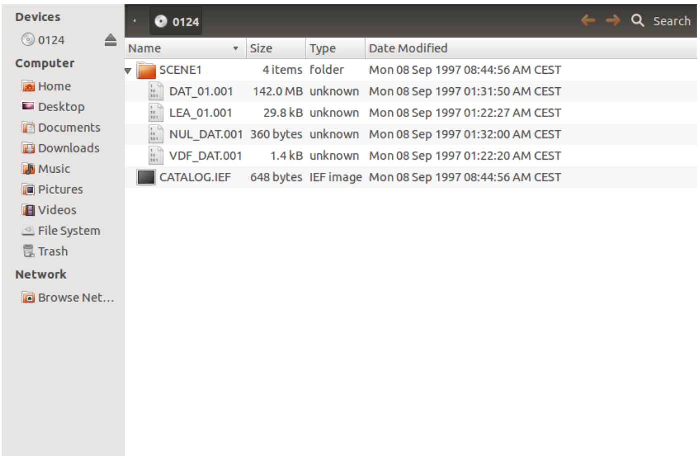


Figure 24: ERS CEOS data structure

V.4.1 Step 1.1:

Create a text file for the reader routine (fig. 19).

Open a terminal and type the following command:

`ERSDataReader /Home/dd/DATA/InSARDemo/Landers/reader.txt -create`

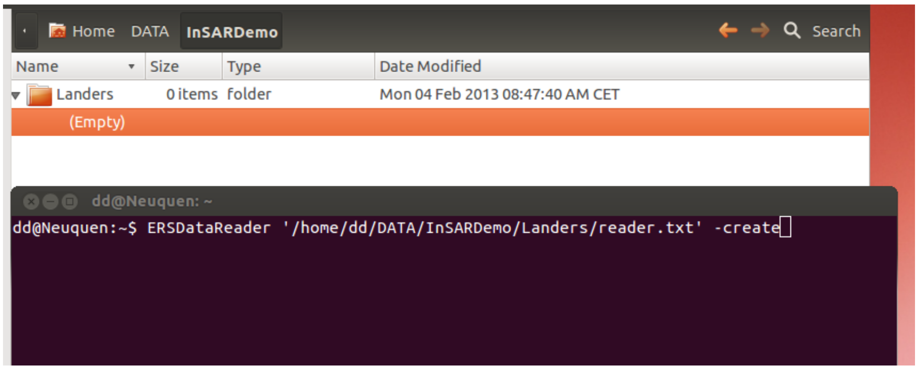


Figure 25: Create reader interface

An example text file named reader.txt will be created at the given location (fig. 20).

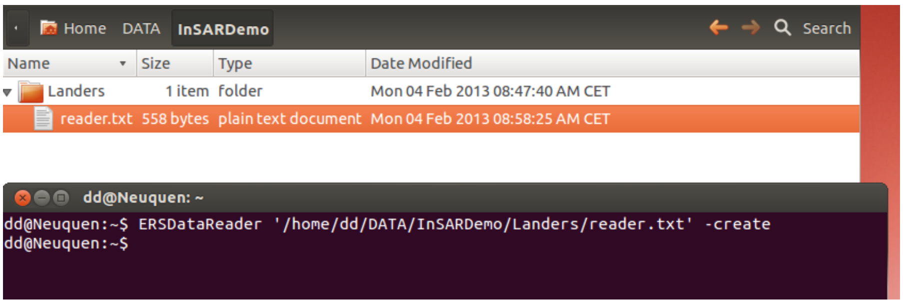


Figure 26: ERSDataReader text interface creation

V.4.2 Step 1.2:

Open the created text file and modify it with respect to the provided data structure (fig. 18).

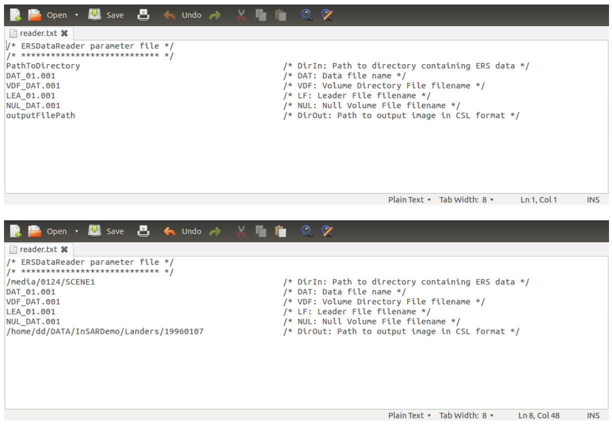


Figure 27: ERSDataReader text interface modification

Figure 21 shows the interface text file to read data acquired on January 7, 1996. We choose to save data at the following location: `/Home/dd/DATA/InSARDemo/Landers/19960107.`

V.4.3 Step 1.3:

Rerun the ERSDataReader command without the –create keyword to read and save data at the requested location.

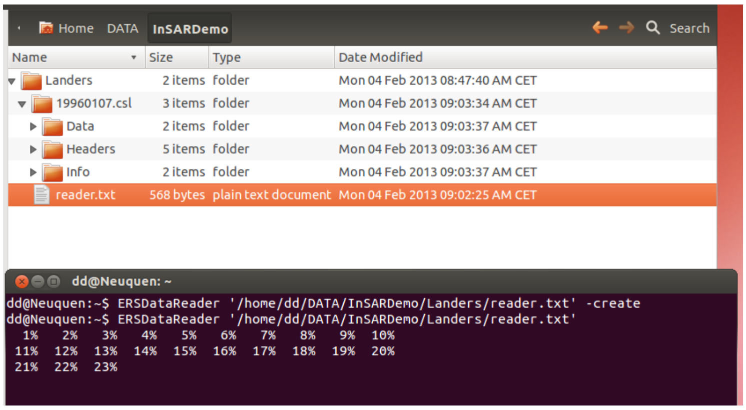


Figure 28: ERS Data Reading and CSL image format

Data is read and saved in CSL format. The CSL format is simply a predefined directory structure into which the data is saved in line with required text files for subsequent processing (fig. 7).

V.4.4 Step1.4:

Update the reader interface text file to read the three other images.

V.5 Step2: InSAR initialization

V.5.1 Step 2.1:

InSAR initialization works in the same way. First, create an interface text file using the keword –`create` in the command line (fig. 23).

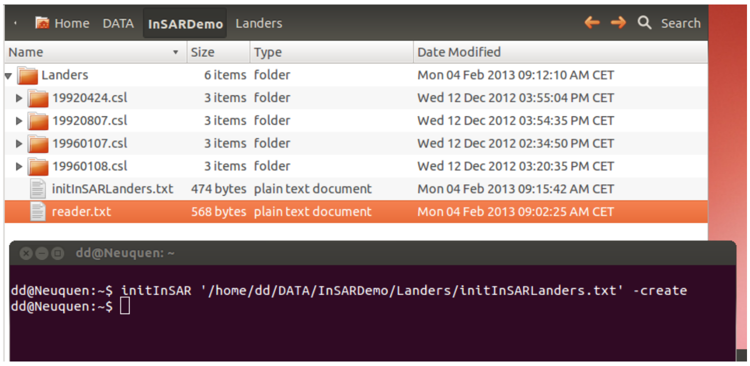


Figure 29: InitInSAR interface text file creation

V.5.2 Step 2.2:

Update the created file choosing the image pair you want to process. Choose a destination location (fig. 24). If the chosen location does not exist, the directory and its subdirectories will be created (fig. 25).

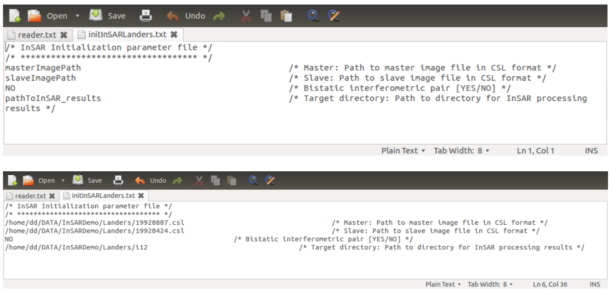


Figure 30: initInSAR text interface modification

V.5.3 Step 2.3:

Rerun the `initInSAR` command without the `–create` keyword (fig 25).

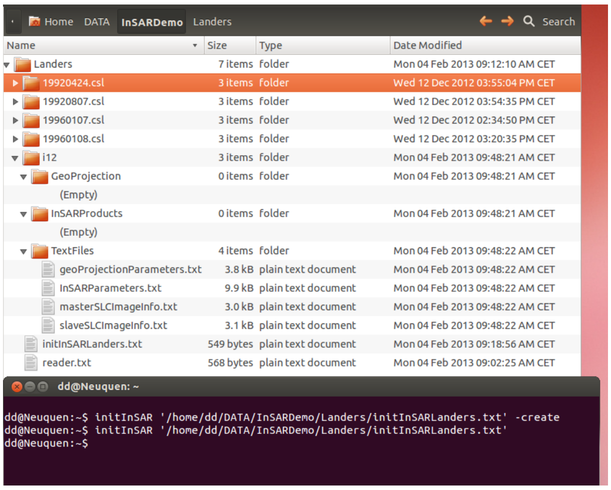


Figure 31: initInSAR - Directory structure

The initInSAR routine will create the required directory structure for subsequent processing. The `./TextFiles` subdirectory contains the info text files issued from each image that are required to perform InSAR processing.

The files `InSARParameters.txt` and `geoProjectionParameters.txt` are the interface text files for the InSAR processing and the geoprojection processing.

The `InSARParameters.txt` is created with default parameters for a classical InSAR processing of ERS or EnviSAT data. In the following, there is no need to modify them. They are fully suitable for the whole processing. However, feel free to modify them at will to evaluate their influence on the processing results and quality.

Predefined names and locations for intermediate and final results can for sure be modified at will.

V.5.4 Step 2.4:

Update the initInSAR text interface to initialize processing for pair 13 and for the topographic pair 34.

V.6 Step 3: coregistration

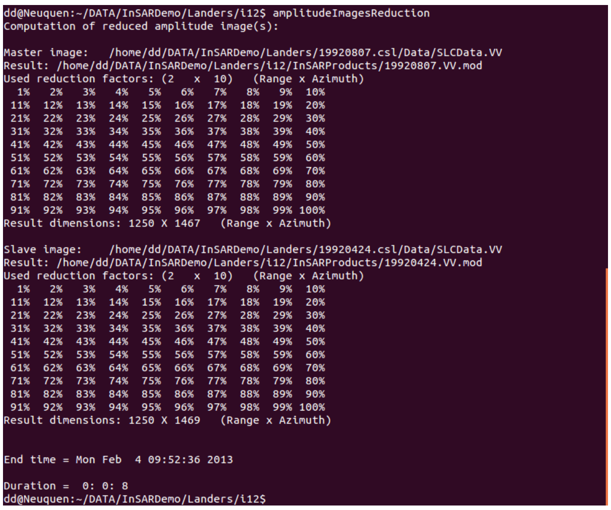
Coregistration is performed in three sub-steps. First, amplitude images must be created. These ones are computed by box averaging of the full resolution complex images. The default reduction factors are 2 pixels in range times 10 pixels in azimuth.

After amplitude image reduction, coarse coregistration followed by fine coregistration can take place.

To run the commands, one can either use the full command name followed by the path to the InSAR parameters file or change the working directory to the created InSAR directory. In this latter case, the user can simply type the command. The corresponding InSAR parameters file will be found in the sub-directory hierarchy.

V.6.1 Step 3.1:

Open a terminal, change your working directory to the displacement pair 12 directory and type the command `amplitudeImageReduction`. At the end of the process, the location and dimension of the amplitude images will be given (fig. 26). These values are also saved in the updated InSAR parameters text file.



```
dd@Neuquen:~/DATA/InSARDemo/Landers/i12$ amplitudeImagesReduction
Computation of reduced amplitude image(s):

Master image:   /home/dd/DATA/InSARDemo/Landers/19920807.csl/Data/SLCData.VV
Result: /home/dd/DATA/InSARDemo/Landers/i12/InSARProducts/19920807.VV.mod
Used reduction factors: (2   x  10)   (Range x Azimuth)
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%
Result dimensions: 1250 X 1467   (Range x Azimuth)

Slave image:    /home/dd/DATA/InSARDemo/Landers/19920424.csl/Data/SLCData.VV
Result: /home/dd/DATA/InSARDemo/Landers/i12/InSARProducts/19920424.VV.mod
Used reduction factors: (2   x  10)   (Range x Azimuth)
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%
Result dimensions: 1250 X 1469   (Range x Azimuth)


End time = Mon Feb  4 09:52:36 2013

Duration =  0: 0: 8
dd@Neuquen:~/DATA/InSARDemo/Landers/i12$
```

Figure 32: Amplitude image reduction process

By default, amplitude images are saved within the `./InSARProducts` subdirectory (fig. 27).

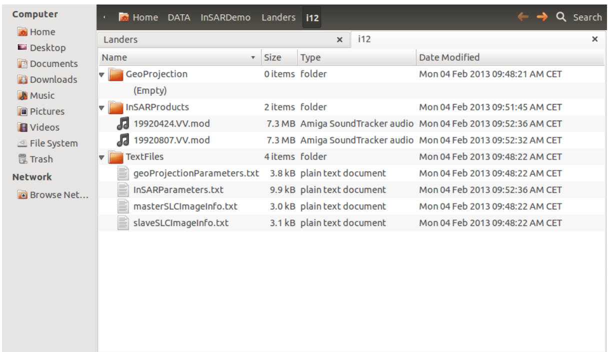


Figure 33: First generated InSAR products – Amplitude images

The created images can be opened using the ImageJ free software. This one can be downloaded at: rsbweb.nih.gov/ij/download.html

To open the data with ImageJ, choose the menu File → Import → Raw (fig. 28).

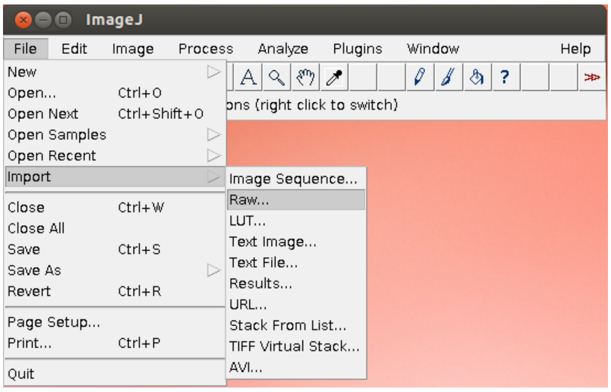


Figure 34: Import image with ImageJ

Select the image to be opened and give the correct dimension of the image. Image type is 32-bit real (float values). With respect to the machine you are using check or un-check the “Little-endian byte order” box (fig. 29).

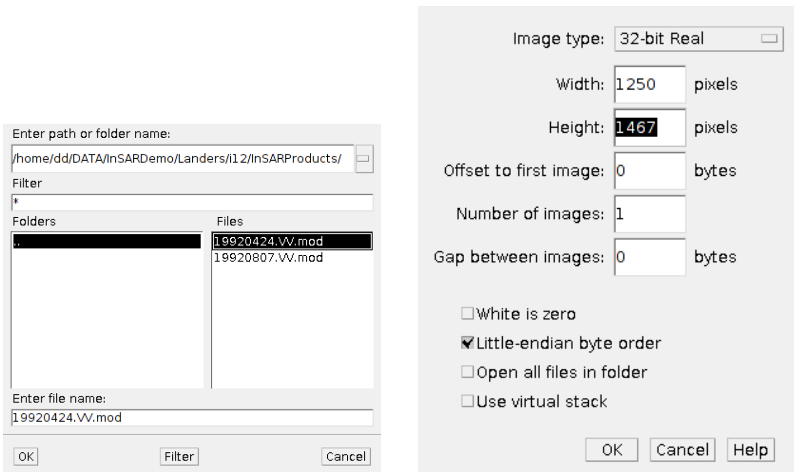


Figure 35: ImageJ data selection and import parameters

Click ok to open the image. ImageJ allows performing some not so basic processing. For example, you can compute the log of the values, which in our case will corresponds to displaying values proportional to the backscattering coefficient (fig. 30).

Use the Image→Adjust→Brightness/Contrast menu to display your image conveniently.

You can now save it in a more convenient image format.

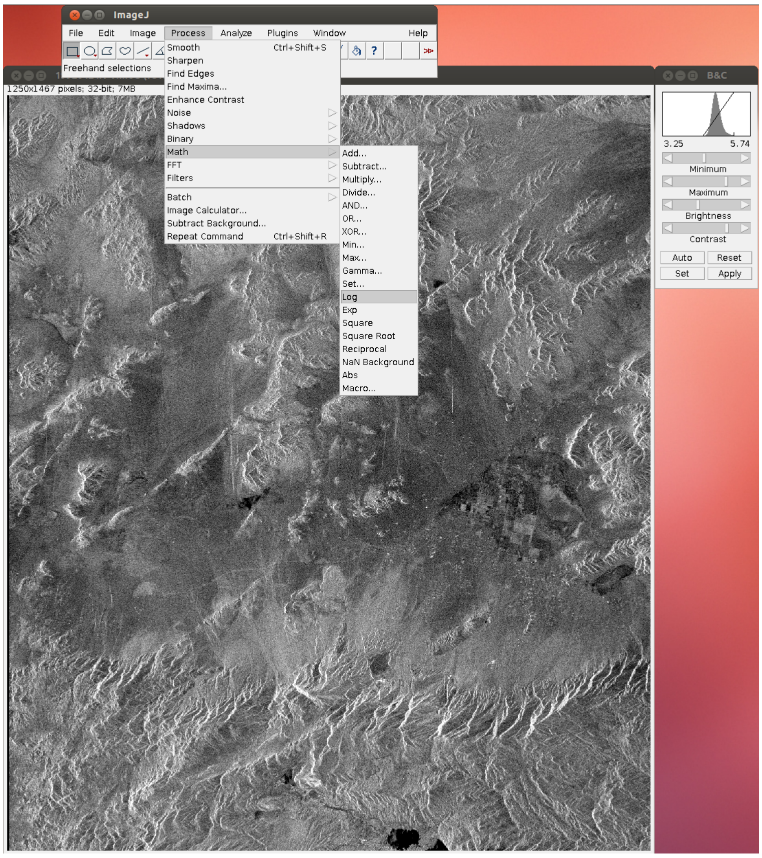


Figure 36: Amplitude image of the Landers site

V.6.2 Step 3.2:

To perform a coarse coregistration, simply type command `coarseCoregistration`.

Process will be performed on windows of 64x64 pixels cut out of the master and slave images. Coregistration values are found by correlation of these windows. Windows are cut out every 100 pixels by default. This value determine the number of candidate anchor points.

At the end of the process, the affine transform linking the slave to the master image is computed and saved in the InSAR parameters text file. The command can be run iteratively to refine the result.

```
Half With at Half Maximum threshold of the correlation peak: 5.000000 x 5.000000 (range x az
imuth pixels)
Correlation window size: 64 x 64 ((range x azimuth pixels)
Window shift: 100 x 100 ((range x azimuth pixels)

============================ PROCESS ===================================
Number of candidate anchor points: 14 X 12 = 168
Line 0; Number of found anchor points: 8         Correlation maximum = 0.722099
Line 1; Number of found anchor points: 6         Correlation maximum = 0.728317
Line 2; Number of found anchor points: 10        Correlation maximum = 0.727071
Line 3; Number of found anchor points: 9         Correlation maximum = 0.797098
Line 4; Number of found anchor points: 6         Correlation maximum = 0.722551
Line 5; Number of found anchor points: 4         Correlation maximum = 0.726472
Line 6; Number of found anchor points: 6         Correlation maximum = 0.762500
Line 7; Number of found anchor points: 8         Correlation maximum = 0.752995
Line 8; Number of found anchor points: 3         Correlation maximum = 0.815495
Line 9; Number of found anchor points: 3         Correlation maximum = 0.687571
Line 10; Number of found anchor points: 8        Correlation maximum = 0.660354
Line 11; Number of found anchor points: 12       Correlation maximum = 0.816893
Line 12; Number of found anchor points: 11       Correlation maximum = 0.695926
Line 13; Number of found anchor points: 11       Correlation maximum = 0.735514

Coregistration results:
*************************
Total number of anchor points: 105 .
Found affine transform:
x2 = Ax x1 + Bx y1 + Cx
 Ax = 1.000000000000002         Bx = 0.000000000000000          Cx = 13.999999999995131

y2 = Ay x1 + By y1 + Cy
 Ay = 0.000608988230158         By = 1.000011090027725          Cy = -103.442435232830789

sigmaRange = 0.000009   sigmaAzimuth = 1.037069 sigmaRangeAzimuth = 1.037069

End time = Mon Feb  4 10:11:54 2013

Duration =  0: 0: 1
dd@Neuquen:~/DATA/InSARDemo/Landers/i12$
```

Figure 37: Coarse coregistration

V.6.3 Step 3.3:

Fine coregistration is performed in exactly the same way. In this case, correlation is performed using windows cut out in the full resolution complexe images and coregistration values are found by local optimization of the coherence. Default window size is 5 X 25 pixels. Default window gap is 20 pixels in both directions. These values lead to a huge number of candidate anchor points. Only those leading to a coherence level higher than a predefined threshold of 0.5 will be retained as valid anchor point for the affine transform calculation (fig. 32).

Just type `fineCoregistration` at the command line.

```
Line 688; Number of found anchor points: 0       Maximum coherence = 0.393093
Line 689; Number of found anchor points: 1       Maximum coherence = 0.495896
Line 690; Number of found anchor points: 0       Maximum coherence = 0.405895
Line 691; Number of found anchor points: 0       Maximum coherence = 0.405272
Line 692; Number of found anchor points: 0       Maximum coherence = 0.407841
Line 693; Number of found anchor points: 0       Maximum coherence = 0.377309
Line 694; Number of found anchor points: 1       Maximum coherence = 0.451793
Line 695; Number of found anchor points: 0       Maximum coherence = 0.401394

Coregistration results:
*************************
Total number of anchor points: 917 .
Found affine transform:
x2 = Ax x1 + Bx y1 + Cx
 Ax = 0.999005755128633         Bx = 0.000002132651958          Cx = -2.233370950891564

y2 = Ay x1 + By y1 + Cy
 Ay = -0.000472731559774        By = 1.000059155790624          Cy = 182.091490714353881

sigmaRange = 0.364869   sigmaAzimuth = 0.421927 sigmaRangeAzimuth = 0.557810

End time = Mon Feb  4 10:59:00 2013

Duration =  0: 0:25
dd@Neuquen:~/DATA/InSARDemo/Landers/i13$
```

Figure 38: Fine coregistration of pair 13

V.6.4 Step 3.4:

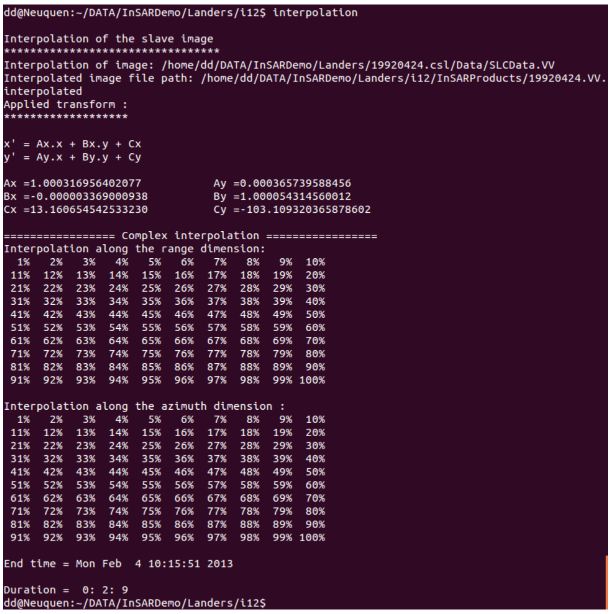
Reiterate these three steps for pair 13 and pair 34. Do not forget to change your working directory to the one of the pair you want to process or use the full path to the corresponding InSAR parameters file as command line parameter for `amplitudeImageReduction`, `coarseCoregistration` and `fineCoregistration` commands.

V.7 Step 4: Interpolation

After each co-registration, the corresponding affine transform is updated in the respective InSAR parameter file. This affine transform is the one that should be applied to the slave image to become superimposable to its master one. So, the affine transform obtained for pair 12 can be applied to image 2 as such to become superimposable to image 1, which is its master one but also the global mater image.

V.7.1 Step 4.1:

To perform interpolation of image 2, simply change your working directory to the one of pair 12 and type the command `interpolation` (fig. 33).



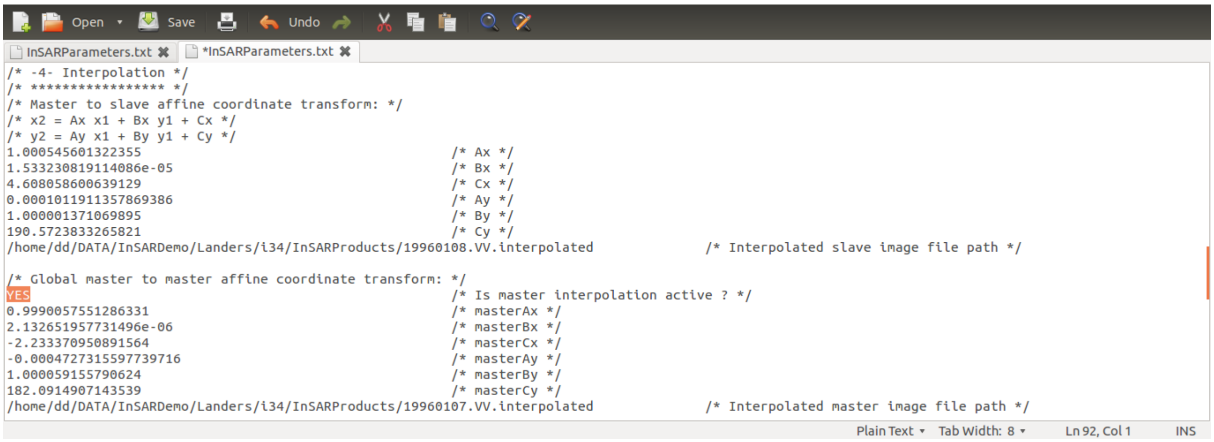
```
dd@Neuquen:~/DATA/InSARDemo/Landers/i12$ interpolation

Interpolation of the slave image
*********************************
Interpolation of image: /home/dd/DATA/InSARDemo/Landers/19920424.csl/Data/SLCData.VV
Interpolated image file path: /home/dd/DATA/InSARDemo/Landers/i12/InSARProducts/19920424.VV.
interpolated
Applied transform :
*******************

x' = Ax.x + Bx.y + Cx
y' = Ay.x + By.y + Cy

Ax =1.000316956402077           Ay =0.000365739588456
Bx =-0.000003369000938          By =1.000054314560012
Cx =13.160654542533230          Cy =-103.109320365878602

================= Complex interpolation =================
Interpolation along the range dimension:
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%

Interpolation along the azimuth dimension :
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%

End time = Mon Feb  4 10:15:51 2013

Duration =  0: 2: 9
dd@Neuquen:~/DATA/InSARDemo/Landers/i12$
```

Figure 39: Pair 12 interpolation

V.7.2 Step 4.2:

With respect to pair 34, in order to have final products, i.e. the topographic phase, in the same geometry than the global master image, two interpolations must take place. Image 3 must be interpolated with respect to the transform found by fine co-registration of pair 13. Then, image 4 must be interpolated using the same transform in addition to the one found by fine co-registration of pair 34 aiming at superimposing image 4 to image 3.

Both interpolations will be performed using the same routine provided the parameter file is modified as follows. The transform found for pair 13 (fig. 32) and save in the corresponding InSAR parameters file must be copied and pasted in the InSAR parameters file of pair 34 in section named “Global master to master affine coordinate transform” (fig. 35) and the flag “Is master interpolation active ?” MUST be turned on setting it to “YES”.



```
/* -4- Interpolation */
/* ***************** */
/* Master to slave affine coordinate transform: */
/* x2 = Ax x1 + Bx y1 + Cx */
/* y2 = Ay x1 + By y1 + Cy */
1.000545601322355                                       /* Ax */
1.533230819114086e-05                                   /* Bx */
4.608058600639129                                       /* Cx */
0.0001011911357869386                                   /* Ay */
1.000001371069895                                       /* By */
190.5723833265821                                       /* Cy */
/home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/19960108.VV.interpolated              /* Interpolated slave image file path */

/* Global master to master affine coordinate transform: */
YES                                                     /* Is master interpolation active ? */
0.9990057551286331                                      /* masterAx */
2.132651957731496e-06                                   /* masterBx */
-2.233370950891564                                      /* masterCx */
-0.0004727315597739716                                  /* masterAy */
1.000059155790624                                       /* masterBy */
182.0914907143539                                       /* masterCy */
/home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/19960107.VV.interpolated              /* Interpolated master image file path */
```

Figure 40: Interpolation parameters modification to perform a double interpolation of both master and slave images

Once this done, interpolation can be performed launching simply the command `interpolation`. Both interpolations will take place in sequence (fig. 35).

The transform effectively applied to image 4 is the combination of both transforms, i.e. the one linking the master to the global master and the one linking the slave to the master in pair 34.

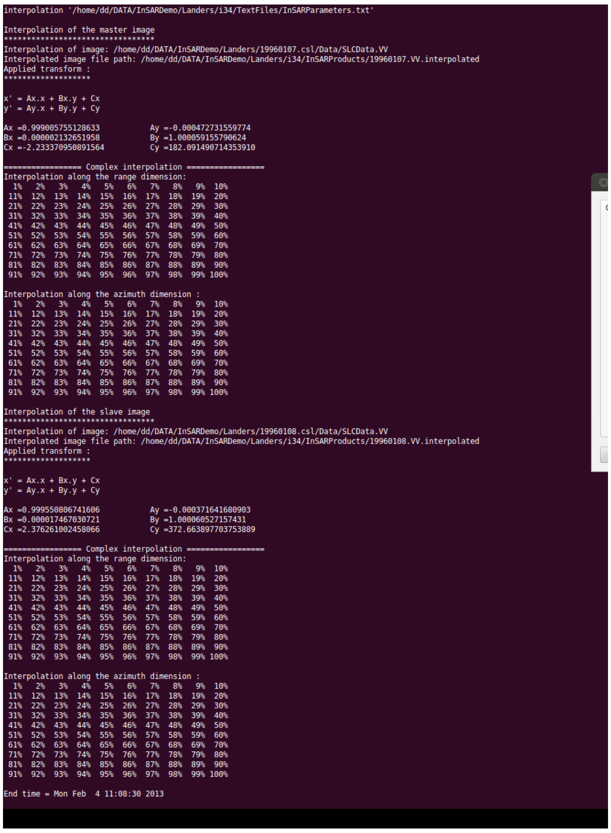


Figure 41: Pair 34 double interpolation

After this step, all images are now co-registered to the global master one. Interferometric products can now be computed in the same slant range Azimuth geometry.

V.8 Step 5: Interferometric product computation

V.8.1 Step 5.1:

Before computing the interferometric products, it is to say the interferograms and coherences images, we have to determine the common area of interest. Each pair has an intersection area that is given in the "Area of interest" section of each InSAR parameters file. This AoI is given with respect to the master image frame and determine the rectangular area common to both acquisitions of the pair under concern.

The user have thus to open the three InSAR parameters files (for pair 12, 13 and 34) and determine the common intersection (fig. 36). Then this common intersection must be pasted in the InSAR parameters files of pair 12 and 34. Doing this, the interferometric products will be computed for the same area of interest defined in the global master frame. If willing to perform processing on a smaller area, just modify the AoI parameters correspondingly.

Pair 12

```
/* Area of interest */
/* *************** */
104                                    /* Lower left azimuth coordinate */
0
14677
2485
```

Pair 13

```
/* Area of interest */
/* *************** */
0                                      /* Lower left azimuth coordinate */
3                                      /* Lower left range coordinate */
13948
2499
```

Pair 34

```
/* Area of interest */
/* *************** */
0                                      /* Lower left azimuth coordinate */
3                                      /* Lower left range coordinate */
13787                                  /* Upper right azimuth coordinate */
2497                                   /* Upper right range coordinate */
```

Common area of interest

```
/* Area of interest */
/* *************** */
104                                    /* Lower left azimuth coordinate */
3                                      /* Lower left range coordinate */
13787                                  /* Upper right azimuth coordinate */
2485                                   /* Upper right range coordinate */
```

Figure 42: Common AoI determination

V.8.2 Step 5.2:

Once the common AoI determined and set into each InSAR parameters file, interferometric products can be computed using command `InSARProductGeneration` (fig. 37).

Pay attention, if you change some parameters influencing the InSAR product computation in one parameter file, you have to change them in the other to get final results having the same dimensions.

```
dd@Neuquen:~/DATA/InSARDemo/Landers/i13$ InSARProductsGeneration '/home/dd/DATA/InSARDemo/Landers/i12/TextFiles/InSARParameters.txt'

Interferometric product computation:

Master image/home/dd/DATA/InSARDemo/Landers/19920807.csl/Data/SLCData.VV
Slave image/home/dd/DATA/InSARDemo/Landers/i12/InSARProducts/19920424.VV.interpolated

Used reduction factors (range x azimuth): 2 X 10
Estimator window size: 2 X 10
Size of interferometric product images: 1239 X 1367
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%

End time = Mon Feb  4 11:17:40 2013

Duration =  0: 1:30
dd@Neuquen:~/DATA/InSARDemo/Landers/i13$ InSARProductsGeneration '/home/dd/DATA/InSARDemo/Landers/i34/TextFiles/InSARParameters.txt'

Interferometric product computation:

Master image/home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/19960107.VV.interpolated
Slave image/home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/19960108.VV.interpolated

Used reduction factors (range x azimuth): 2 X 10
Estimator window size: 2 X 10
Size of interferometric product images: 1239 X 1367
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%

End time = Mon Feb  4 11:19:47 2013

Duration =  0: 1:48
dd@Neuquen:~/DATA/InSARDemo/Landers/i13$
```

Figure 43: InSAR product generation

As can be seen on figure 37, the full path to the InSAR parameters file to be considered has been used as command line parameter. We also observe that, since the flag for "Is master interpolation active" is set to "YES" for pair 34, it is well both interpolated images that are used to generate the interferometric products in the geometry of the global master image.

Dimensions off all products are identical and saved in the updated InSAR parameters files.

Figure 38 shows both obtained interferograms. The left one is issued from pair 12 and contains the displacement phase. The right interferogram is the one issued from pair 34 and is supposed to contain only the topographic phase.

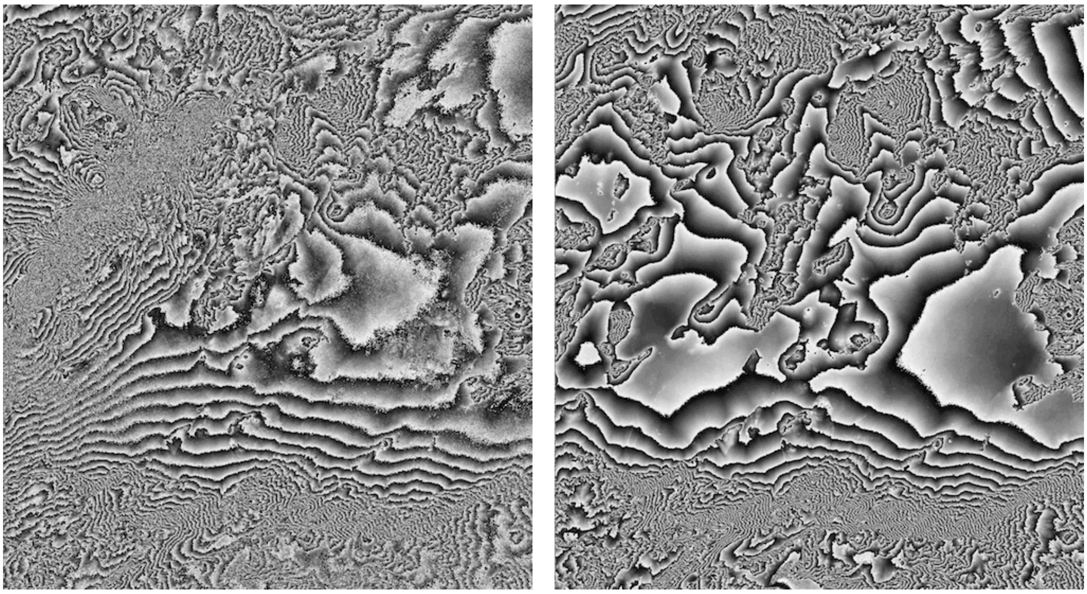


Figure 44: Landers area interferograms

V.9 Step 6: Phase unwrapping

Interferogram of pair 34 must now be unwrapped to generate a topographic reference that can be converted in to a topographic phase and removed from interferogram of pair 12.

Two techniques of phase unwrapping are implemented. The first one is a deterministic phase unwrapping allowing to get a very good first phase component on the whole scene. The second phase unwrapping technique is a classical method based on residue connection algorithmic. It is only this one that is described here.

V.9.1 Step 6.1:

Phase unwrapping is divided in four steps. The first step consists simply in locating the residues. To perform this step, first change the working directory to the one of pair 34, and then enter the `residuesSearch` command (fig. 39).

V.9.2 Step 6.2:

The second step consists in computing a biased coherence image that will be used to guide the residue connection process. Process is launched using the `biasedCoherenceEstimation` command (fig.39).

V.9.3 Step 6.3:

The third step consists in connecting the residues. Process is launched using the `residuesConnexion` command (fig. 39).

V.9.4 Step 6.4:

The last step is the phase unwrapping process itself (fig. 39). The connection map will be read and the interferometric phase will integrated following a path that rounds the connections. The unwrapped phase is then converted in height, taking into account the orbitography and the acquisition geometry of the interferometric pair.

```
dd@Neuquen:~/DATA/InSARDemo/Landers/i13$ cd '/home/dd/DATA/InSARDemo/Landers/i34'
dd@Neuquen:~/DATA/InSARDemo/Landers/i34$ residuesSearch
Residues search
**************
Interferometric phase: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/interfero.VV-VV
Residue image file : /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/residus.VV-VV
Result file dimensions: 1230 X 1358
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%
Nombre of found residues: 16202

End time = Mon Feb  4 11:24:46 2013

Duration =  0: 0: 1
dd@Neuquen:~/DATA/InSARDemo/Landers/i34$ biasedCoherenceEstimation
Biased coherence estimation:
****************************
Master amplitude image: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/19960107.VV.mod
Slave amplitude image: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/19960108.VV.mod
Interférogramme: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/interfero.VV-VV
Filtered interférogram: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/interfero.f1x5.VV-VV
Biased coherence image: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/biasedCoherence.VV-VV
Biased Coherence image size: 1231 X 1359
Estimator window size: 5 X 5
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%


End time = Mon Feb  4 11:24:55 2013

Duration =  0: 0: 1
dd@Neuquen:~/DATA/InSARDemo/Landers/i34$ residuesConnexion
Residues connexion
******************
Biased coherence file/home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/biasedCoherence.VV-VV
Residues file/home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/residus.VV-VV
Cleaning coherence threshold: 0.250000
False residues coherence threshold: 0.100000
Coherence scale: 0 --> 1000
Residues still to be connected:
1     0.060000
Connexion image file: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/connexions.VV-VV
Dimensions: 1230 X 1358

End time = Mon Feb  4 11:25:06 2013

Duration =  0: 0: 3
dd@Neuquen:~/DATA/InSARDemo/Landers/i34$ phaseUnwrapping
Phase unwrapping
****************
interferogram file: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/interfero.VV-VV
Connexions image file: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/connexions.VV-VV
Unwrapped phase file: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/unwrappedPhase.VV-VV
Slant range DEM file: /home/dd/DATA/InSARDemo/Landers/i34/InSARProducts/slantRangeDEM.VV-VV
Ambiguity altitude at interferogram centre: 67.342120
Phase unwrapping done
Unwrapped phase image size: 1229 X 1357

Phase to height conversion
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%

End time = Mon Feb  4 11:25:53 2013

Duration =  0: 0:12
dd@Neuquen:~/DATA/InSARDemo/Landers/i34$
```

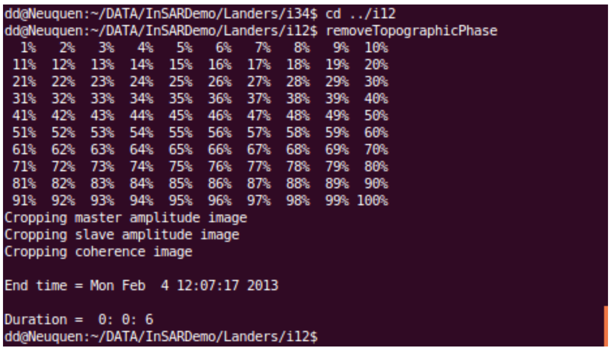
Figure 45: Phase unwrapping sequence

In this last step, the global master to master transform is taken into account for proper phase to height conversion. The net result is a topographic map in the slant range geometry of the global master image.

V.10 Step 7: DInSAR

Now that we have a reference topography in the correct geometry, we just have to indicate it within the InSAR parameters file of pair 12. Just copy-past the 3 last lines of the InSAR parameter file of pair 34 to the InSAR parameters file of pair 12 indicating the location of slant range DEM (field: `/* Slant range DEM file path */`) and its dimensions.

DInSAR process is simply conducted launching the command `removeTopographicPhase` (fig. 40).



```
dd@Neuquen:~/DATA/InSARDemo/Landers/i34$ cd ../i12
dd@Neuquen:~/DATA/InSARDemo/Landers/i12$ removeTopographicPhase
  1%   2%   3%   4%   5%   6%   7%   8%   9%  10%
 11%  12%  13%  14%  15%  16%  17%  18%  19%  20%
 21%  22%  23%  24%  25%  26%  27%  28%  29%  30%
 31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
 41%  42%  43%  44%  45%  46%  47%  48%  49%  50%
 51%  52%  53%  54%  55%  56%  57%  58%  59%  60%
 61%  62%  63%  64%  65%  66%  67%  68%  69%  70%
 71%  72%  73%  74%  75%  76%  77%  78%  79%  80%
 81%  82%  83%  84%  85%  86%  87%  88%  89%  90%
 91%  92%  93%  94%  95%  96%  97%  98%  99% 100%
Cropping master amplitude image
Cropping slave amplitude image
Cropping coherence image

End time = Mon Feb  4 12:07:17 2013

Duration =  0: 0: 6
dd@Neuquen:~/DATA/InSARDemo/Landers/i12$
```

Figure 46: Topographic phase removal

A flattened interferogram will be generated and save with the same name than the interferogram, adding the extension ".differential" (fig. 41).

All other products will be cropped to have the same final dimensions.

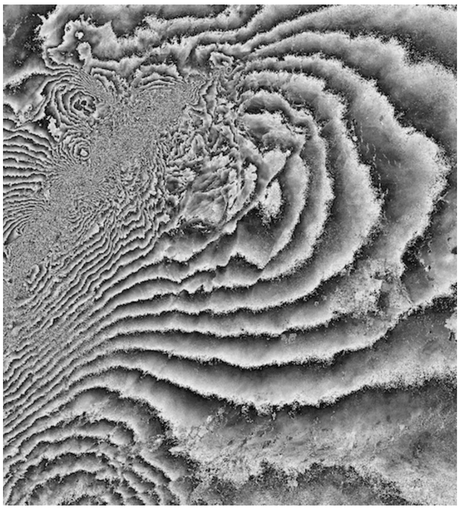


Figure 47: Differential interferogram

As can be seen on figure 41, the differential interferogram appears flipped with respect to what we are used to see within publications. This is simply due to the fact we do not have made any geometrical manipulation of the result and display it as read by ImageJ. The origin of the image is the upper left corner. Since used acquisitions are descending, west is on the right.

V.11 Step 8: Geoprojection

Products can then be geoprojected on the reference topography using the `geoProjection` command.

In our case, we have chosen to geoproject in UTM on a grid of 50x50 meters. The geoprojection routine only projects the DEM itself, the amplitude images and the coherence image. If willing to geoproject the interferogram, you have to "trick" the software, changing into the InSARParameters.txt text file, the location of an amplitude image or of the coherence image with the one of the interferogram you want to project.

Projected interferogram is shown on figure 42.

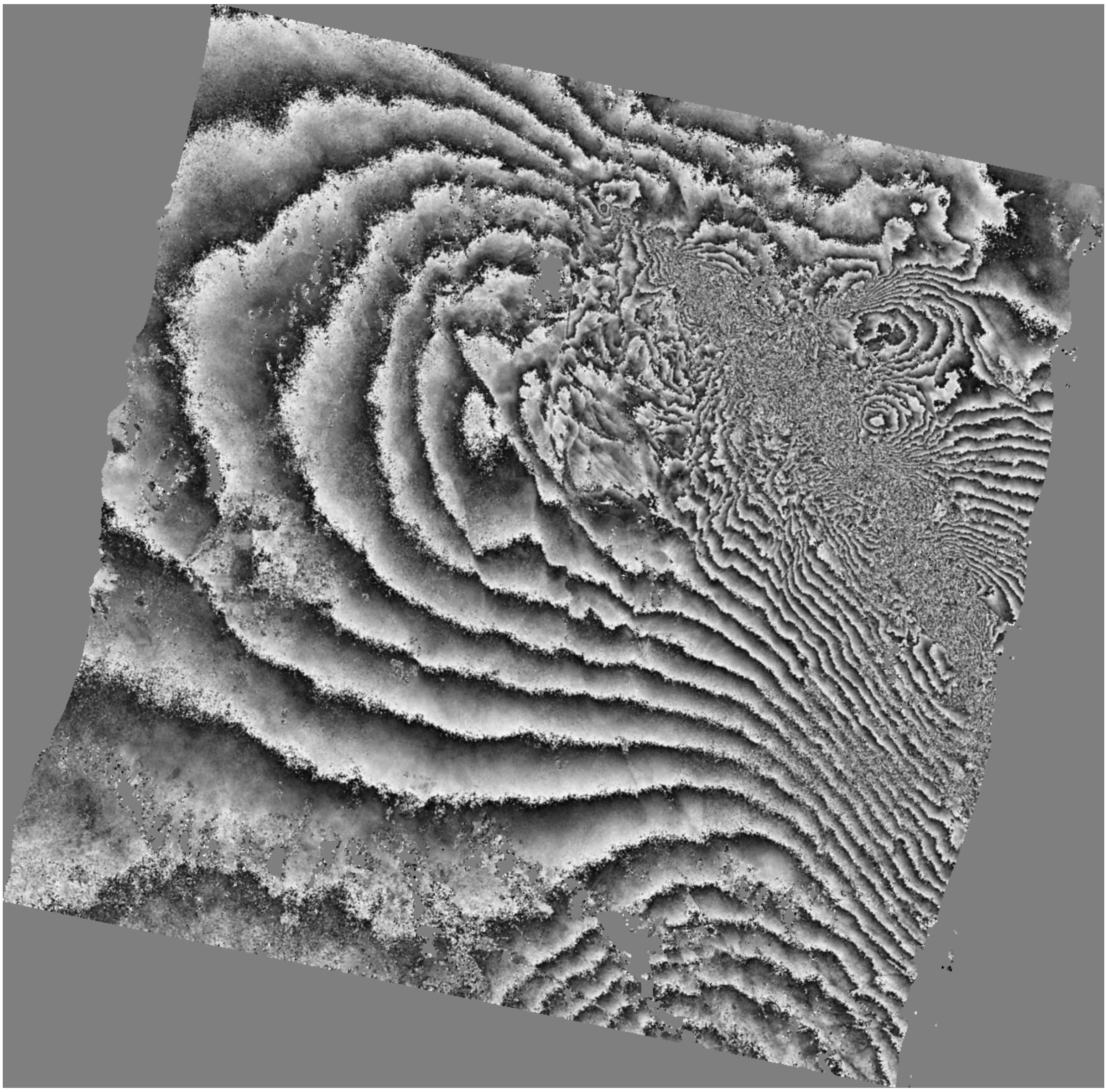


Figure 48: Projected differential interferogram

VI. CSK SLC image calibration

A specific routine allowing to calibrate CSK Single Look Complex data is provided within the suite.

This routine name SLC2MDG uses its specific parameter text file and allow performing full calibration of the data including resampling and projection in ground range (MDG), on the flat Earth ellipsoid (GEC) or on a provided ellipsoid (GTC).

The acronyms MDG (Multilook Detected Ground), GEC (Geocoded Ellipsoid Corrected) and GTC Geocoded Terrain Corrected) are those used in the CosmoSkymed nomenclature.

VI.1 Specific CSK data reading

Once a level 1A CSK product is read using the `CSKDataReader` routine, all the required information needed to perform the calibration are extracted from the HDF5 data structure and saved within the ./Info directory of the image in CSL format.

A first parameter file named `SLCCalibrationInfo.txt` is created. Next figure shows an example in the case of a dual polarization CSK image.

SLCCalibrationInfo.txt

```
/* SLC calibration info text file */
/* ****************************** */
0.68                                   /* Azimuth Hamming window coefficient */
133.3910675048828                      /* Rescaling factor */
NO                                     /* Is calibration constant compensation performed */

YES                                    /* Is range spreading loss correction performed? */
YES                                    /* Is range spreading loss normalization performed? */
780000                                 /* Reference slant range [m] */
2                                      /* Reference slant range exponent */

YES                                    /* Is incidence angle correction performed? */
YES                                    /* Is incidence angle normalization performed? */
40                                     /* Reference incidence angle [deg] */

2                                      /* Number of data layers */
6.67103675298199e+25                   /* VV calibration constant */
YES                                    /* Is VV elevation antenna pattern correction performed? */
37.78                                  /* VV beam centre off-nadir angle [deg] */
-6                                     /* VV elevation antenna pattern origin with respect to beam
centre [deg] */
0.01999999999999957                    /* VV elevation antenna pattern sampling [deg] */
VV_TwoWayElevationAntennaPattern.txt   /* VV elevation antenna pattern file name */
601                                    /* VV elevation antenna pattern number of samples */

6.82642516057392e+25                   /* VH calibration constant */
YES                                    /* Is VH elevation antenna pattern correction performed? */
37.78                                  /* VH beam centre off-nadir angle [deg] */
-6                                     /* VH elevation antenna pattern origin with respect to beam
centre [deg] */
0.01999999999999957                    /* VH elevation antenna pattern sampling [deg] */
VH_TwoWayElevationAntennaPattern.txt   /* VH elevation antenna pattern file name */
601                                    /* VH elevation antenna pattern number of samples */
```

Figure 49: SCLCalibrationInfo.txt example

This file gives all the required parameters to complete the calibration.

Two-way elevation antenna pattern is saved in a separated file. If several antenna patterns are made available, as it is the case in the dual-polarization acquisition scheme, they are saved separately.

VI.2 Calibration parameter file

An example calibration parameter file can be created as usual using the "`-create`" keyword in the command line giving the targeted file path as first parameter of the `SLC2MDG` routine (figure 50).

The first parameter is the image location in CSL format that must be calibrated.

Second parameter is the path to the calibrated result. If set to the value "`Default`", result will be saved within the CSL image container under the name MDGData.XX, GECData.XX or GTCDATA.XX depending on the chosen projection (where XX is the polarization scheme). A corresponding txt file will be saved within the ./Info subdirectory of the image.

Noise reduction can be obtained through either multilooking or box averaging. Multilooking has precedence. If set to one look, no multilooking will take place and it is box averaging that will be considered.

It must be noted that in the CSK Image Calibration procedure (see e-geos web site), only box averaging is mentioned.

Next parameter determines if backscattering coefficient is expressed in db or not.

Next section determines the target projection (MDG, GEC or GTC) and the coordinate system (GEC or UTM). The coordinate system is only taken into account if projection is made on the geoid or on an external DEM. To project on an external DEM, the image to be calibrated must contain a slant range DEM within its ./Data subdirectory and a description of the latte in its ./Info subdirectory. See chapter IV to know how to generate such slant range DEM from an external DEM provided in longitude – latitude on WGS84.

Following parameters concern projection characteristics. For each projection, one must define the target sampling in projection unit. For GEC or UTM, an interpolation radius must be given. If set to zero, it will be estimated from the referencing of the slant range image into the destination geometry. Last parameter set define the resampling method (see geo projection chapter for more details).

```
calib.txt
/* Calibration, multilooking and projection of SLC data - parameter file */
/* ********************************************************************** */
/DATA/CSK/Missiones/20120823.csl                          /* Input file path to image in CSL format */
Default                                           /* Output file path (Default = within CSL format) */

/* Multilooking */
/* ************ */
1                                                 /* Number of looks */

/* Box averaging */
/* ************* */
/* Taken into account only if multilooking is not requested (number of looks = 1) */
1                                                 /* X box averaging size */
5                                                 /* Y box averaging size */

/* Detection */
/* ************ */
db                                                /* Backscattering coefficient scale [db or linear] */

/* Projection selection */
/* ******************** */
GTC                                               /* Targeted projection [SLC, MDG, GEC or GTC] (No multilooking if SLC) */
GEC                                               /* Geographical coordinate system [UTM/GEC] (for GEC or GTC projection) */

/* Ground range projection parameters */
/* ********************************** */
20                                                /* Ground range sampling [m] */
20                                                /* Azimuth sampling [m] */

/* Geographical projection parameters (GEC) */
/* ********************************** */
0.000277777                                       /* Longitude sampling [dd] */
0.000277777                                       /* Latitude sampling [dd] */
0.                                        /* Longitude interpolation radius */
0.                                        /* Latitude interpolation radius */

/* UTM projection parameters */
/* ************************* */
20                                        /* Easting sampling [m] */
20                                        /* Northing sampling [m] */
0                                         /* Easting interpolation radius [m] */
0                                         /* Northing interpolation radius [m] */

/* Resampling parameters */
/* ********************* */
TRI                                               /* Resampling method : TRI = Triangulation; AV = weighted average; NN = nearest neighbour */
ID                                                /* Weighting method : ID = inverse distance; LORENTZ = lorentzian */
1.0                                               /* ID smoothing factor */
1.0                                               /* ID weighting exponent */
1.0                                               /* FWHM : Lorentzian Full Width at Half Maximum */
```

Figure 50: Calibration parameter file example

In the case of GEC or GTC projection, a basic .kml file is saved within the ./info directory of the image. This file allows visualizing the geographical location of the calibrated image on Google Earth. Name of the kml file will be GECData.XX.kml or GTCData.XX.kml depending on the projection, where XX is the polarization scheme.

VII. Annexe

InSAR Parameter file example:

```
/* InSAR parameters file */
/* ********************* */

/* Input images */
/* ************ */
/Volumes/Vi-X/TDX/20110623/20110623_TDX.csl             /* Master image file path [CSL image format] */
VV                                                      /* Master polarization channel */
14048                                                   /* Master image range dimension [pixels] */
18072                                                   /* Master image azimuth dimension [rows] */
/Volumes/Vi-X/TDX/20110704/20110704_TDX.csl             /* Slave image file path [CSL image format] */
VV                                                      /* Slave polarization channel */
14048                                                   /* Slave image range dimension [pixels] */
17968                                                   /* Slave image azimuth dimension [rows] */
538664.38                                               /* Master image minimum slant range */
538679.81                                               /* Slave image minimum slant range */
0.90940338                                              /* Slant range sampling [m] */
9.6499999e+09                                           /* Radar carrier frequency [Hz] */
YES                                                     /* Bistatic interferometric pair */

/* Area of interest */
/* **************** */
0                                                       /* Lower left azimuth coordinate */
0                                                       /* Lower left range coordinate */
17803                                                   /* Upper right azimuth coordinate */
13980                                                   /* Upper right range coordinate */

/* Satellite - Earth centre distance */
/* ********************************* */
0.002607886792811633                                    /* Linear variation [m]/[azimuth row] */
6890003.672026199                                       /* Constant term [m] */

/* Baseline parameters */
/* ******************* */
-91.57747783224457                            /* Horizontal baseline component constant term [m] */
-9.800452167162974e-12                        /* Horizontal baseline variation with range position [m]/[range pix] */
-2.066729183703988e-05                        /* Horizontal baseline variation with azimuth position [m]/[azimuth pix] */
34.67007988017943                             /* Vertical baseline component constant term [m] */
-1.903348149557635e-11                        /* Vertical baseline variation with range position [m]/[range pix] */
5.746424011876388e-06                         /* Vertical baseline variation with azimuth position [m]/[azimuth pix] */
80.42713926794413                                       /* Ambiguity altitude at scene centre */

/* ******************* */
/* * InSAR processing * */
/* ******************* */
/* -1- Box averaging  */
/* ***************** */
/DATA/DD/TDX/i12TDX/InSARProducts/20110623_TDX.VV.mod             /* Reduced master amplitude image file path */
0                                                       /* Reduced master amplitude image range dimension [pix] */
0                                                       /* Reduced master amplitude image azimuth dimension [pix] */
/DATA/DD/TDX/i12TDX/InSARProducts/20110704_TDX.VV.mod             /* Reduced slave amplitude image file path */
0                                                       /* Reduced slave amplitude image range dimension [pix] */
0                                                       /* Reduced slave amplitude image azimuth dimension [pix] */
2                                                       /* Range reduction factor [pix] */
10                                                      /* Azimuth reduction factor [pix] */

/* -2- Coarse coregistration */
/* ************************* */
-61                                                     /* Estimated range shift [pix] */
-163                                                    /* Estimated azimuth shift [pix] */
64                                                      /* Coarse coregistration range window size [pix] */
64                                                      /* Coarse coregistration azimuth window size [pix] */
100                                                     /* Coarse registration range distance between anchor points [pix]
*/
100                                                     /* Coarse registration azimuth distance between anchor points
[pix] */
0.5                                                     /* Coarse coregistration correlation threshold */
5                                                       /* Corelation peak range width threshold */
5                                                       /* Corelation peak azimut width threshold */

/* -3- Fine coregistration */
/* *********************** */
5                                                       /* Fine coregistration range window size [pix] */
25                                                      /* Fine coregistration azimuth window size [pix] */
3                                                       /* Estimated range error [pix] */
3                                                       /* Estimated azimuth error [pix] */
20                                                      /* Fine registration range distance between anchor points [pix]
*/
20                                                      /* Fine registration azimuth distance between anchor points [pix]
*/
0.5                                                     /* Fine coregistration correlation threshold */

/* -4- Interpolation */
/* ***************** */
/* Master to slave affine coordinate transform: */
/* x2 = Ax x1 + Bx y1 + Cx */
/* y2 = Ay x1 + By y1 + Cy */
1.00032864706423                                        /* Ax */
1.249877038171203e-05                                   /* Bx */
61.79610115441591                                       /* Cx */
-7.363303352375497e-09                                  /* Ay */
1.000008913883008                                       /* By */
163.5179618746576                                       /* Cy */
/DATA/DD/TDX/i12TDX/InSARProducts/20110704_TDX.VV.interpolated                    /* Interpolated slave image file path */
```

```
/* Global master to master affine coordinate transform: */
NO                                                          /* Is master interpolation active ? */
1                                                           /* masterAx */
0                                                           /* masterBx */
0                                                           /* masterCx */
0                                                           /* masterAy */
1                                                           /* masterBy */
0                                                           /* masterCy */
/DATA/DD/TDX/i12TDX/InSARProducts/20110623_TDX.VV.interpolated                  /* Interpolated master image file path */

/* -5- Interferometric products computation */
/* **************************************** */
0                                                           /* Interferometric products range dimension [pix] */
0                                                           /* Interferometric products azimuth dimension [pix] */

/DATA/DD/TDX/i12TDX/InSARProducts/interfero.VV-VV           /* Interferogram file path */
/DATA/DD/TDX/i12TDX/InSARProducts/interfero.f1x5.VV-VV              /* Filtered interferogram file path */
1                                                           /* Range filter Full Width at Half Maximum */
5                                                           /* Azimuth filter Full Width at Half Maximum */

/DATA/DD/TDX/i12TDX/InSARProducts/coherence.VV-VV           /* Coherence file path */
2                                                           /* Coherence estimator range window size [pix] */
10                                                          /* Coherence estimator azimuth window size [pix] */
5                                                           /* Square spiral size [pix] */

/* -6- Phase unwrapping */
/* ******************** */
/DATA/DD/TDX/i12TDX/InSARProducts/firstPhaseComponent.VV-VV          /* First phase component file path */
-9999                                                       /* First phase component excluding value */
/DATA/DD/TDX/i12TDX/InSARProducts/residualInterferogram.VV-VV                 /* Residual interferogram file path */
/DATA/DD/TDX/i12TDX/InSARProducts/biasedCoherence.VV-VV             /* Biased coherence file path */
0                                                           /* Biased coherence range dimension [pix] */
0                                                           /* Biased coherence azimuth dimension [pix] */
5                                                           /* Biased coherence estimator range window size [pix] */
5                                                           /* Biased coherence estimator azimuth window size [pix] */
5                                                           /* Biased coherence square spiral size [pix] */

/DATA/DD/TDX/i12TDX/InSARProducts/residus.VV-VV             /* Residus image file path */
0                                                           /* Residus image range dimension [pix] */
0                                                           /* Residus image azimuth dimension [pix] */

/DATA/DD/TDX/i12TDX/InSARProducts/connexions.VV-VV                  /* Connexions image file path */
0                                                           /* Connexions image range dimension [pix] */
0                                                           /* Connexions image azimuth dimension [pix] */
2                                                           /* Range minimum search radius [pix] */
2                                                           /* Azimuth minimum search radius [pix] */
1                                                           /* Range search radius step [pix] */
1                                                           /* Azimuth search radius step [pix] */
1000                                                        /* Coherence scaling */
3                                                           /* Connexion process mode */
0.25                                                        /* Coherence cleaning threshold */
0.1                                                         /* False residue coherence threshold */

/DATA/DD/TDX/i12TDX/InSARProducts/unwrappedPhase.VV-VV              /* Unwrapped phase file path */
0                                                           /* Unwrapped phase range dimension [pix] */
0                                                           /* Unwrapped phase azimuth dimension [pix] */

/DATA/DD/TDX/i12TDX/InSARProducts/slantRangeDEM.VV-VV               /* Slant range DEM file path */
0                                                           /* Slant range DEM range dimension [pix] */
0                                                           /* Slant range DEM azimuth dimension [pix] */
```

Geoprojection parameter file example:

```
/* Geoprojection parameters file */
/* ***************************** */

/DATA/DD/TDX/i12TDX/TextFiles/InSARParameters.txt          /* InSAR parameters file */
UTM                                                        /* Chosen projection (UTM or GEC) */

/* Projection ellipsoid */
/* ******************** */
WGS84                                                      /* Ellipsoid ID */
6378137.0                                                  /* Semi major axis */
6356752.314245179295540                                    /* Semi minor axis */
0                                                          /* X0 */
0                                                          /* Y0 */
0                                                          /* Z0 */

/* Geographical projection parameters (GEC) */
/* ********************************** */
0.00055555                                                 /* Longitude sampling [dd] */
0.00055555                                                 /* Latitude sampling [dd] */
0.0007                                                     /* Longitude interpolation radius */
0.0007                                                     /* Latitude interpolation radius */

/* UTM projection parameters */
/* ************************* */
100                                                        /* Easting sampling [m] */
100                                                        /* Northing sampling [m] */
70                                                         /* Easting interpolation radius [m] */
70                                                         /* Northing interpolation radius [m] */

/* Resampling parameters */
/* ********************* */
TRI                                                        /* Resampling method : TRI = Triangulation; AV = weighted
average; NN = nearest neighbour */
ID                                                         /* Weighting method : ID = inverse distance; LORENTZ = lorentzian
*/
1.0                                                        /* ID smoothing factor */
1.0                                                        /* ID weighting exponent */
1.0                                                        /* FWHM : Lorentzian Full Width at Half Maximum */
/pathToMaskFile                                            /* Mask file */
-9999.                                                     /* Excluding height value */
-400.                                                      /* Minimal height value */
8000.                                                      /* Maximal height value */

/* Temporary projection files */
/* **************** */
/DATA/DD/TDX/i12TDX/GeoProjection/slantRangeDEM.VV-VV.xRef            /* X referencing file */
/DATA/DD/TDX/i12TDX/GeoProjection/slantRangeDEM.VV-VV.yRef            /* Y referencing file */
/DATA/DD/TDX/i12TDX/GeoProjection/slantRangeDEM.VV-VV.UTM.100x100.TRI.projMat             /* Projection matrix file */

/* Geoprojected result files */
/* ************************* */
/* To prevent geoprojection of an InSAR product, replace its destination file name by the keyword "NO" */
/DATA/DD/TDX/i12TDX/GeoProjection/slantRangeDEM.VV-VV.UTM.100x100.TRI           /* Projected DEM */
/DATA/DD/TDX/i12TDX/GeoProjection/20110623_TDX.VV.mod.UTM.100x100.TRI           /* Projected master amplitude */
/DATA/DD/TDX/i12TDX/GeoProjection/20110704_TDX.VV.mod.UTM.100x100.TRI           /* Projected slave amplitude */
/DATA/DD/TDX/i12TDX/GeoProjection/coherence.VV-VV.UTM.100x100.TRI               /* Projected coherence image */

/* Geoprojection frame */
/* ******************* */
/* A geoprojection frame can be set in chosen projection units */
/* If values are set to 0., the frame will be computed from the InSAR DEM */
0.                                                         /* xMin */
0.                                                         /* xMax */
0.                                                         /* yMin */
0.                                                         /* yMax */

0                                                          /* X size of geoprojected products [pix] */
0                                                          /* Y size of geoprojected products [pix] */
```

Have fun...