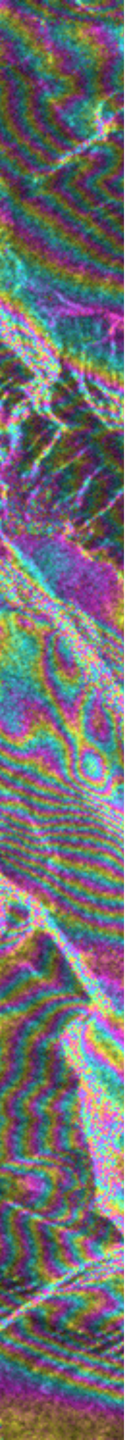


GAMMA REMOTE SENSING

Interferometric processing

The ISP and DIFF&GEO modules

Gamma Remote Sensing AG

Worbstrasse 225

3073 Gümligen

Switzerland

http://www.gamma-rs.ch

Steps of interferometric processing

- Input: 2 SLC images covering the same area (same orbital track)
- Co-registration of SLC images
- Computation of interferogram
- Removal of flat Earth from interferogram
- *Removal of topographic phase from interferogram*

→ *generation of differential interferogram*

- Computation of interferometric coherence
- Phase unwrapping
- *Estimation of displacement*

DInSAR processing steps in italic

Products of InSAR and DInSAR processing

After InSAR processing we obtain the following products

- Interferogram
    - Interferometric Phase + correlation
- Coherence
- Multi-look intensity images
- Unwrapped interferometric phase
- Displacement map

All images are obtained in the radar geometry

Geocoding the images allows the transformation from radar to map geometry

Pre-processing

Why is this done?

Images are on different support media, in specific formats, generally they are not accompanied by all information required for processing.

- Preparation of data
- Co-registration of SLC images
- Computation of interferogram
- Generation of differential interferogram
- Computation of coherence
- Phase unwrapping
- Estimation of displacement

- Data import into processing software
- Improve orbit state vectors (if possible), just as explained for SAR processing
- Prepare calibration files and calibrate images (also possible at the end), just as explained for SAR processing

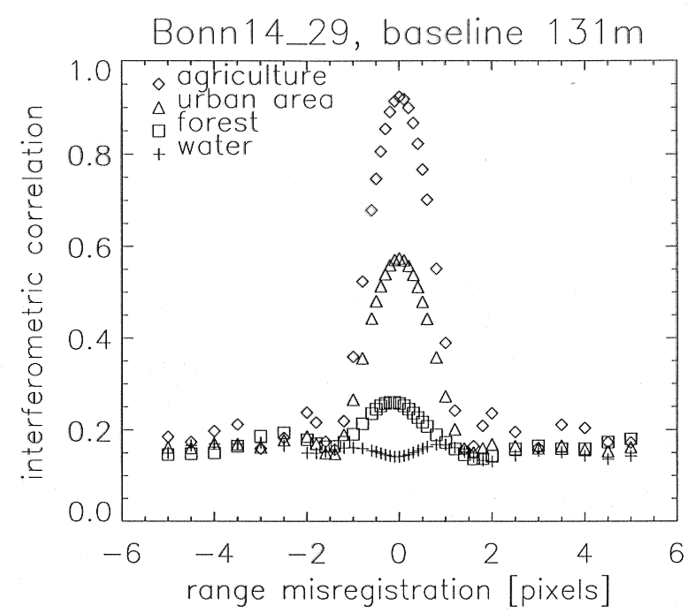
Co-registration

Why is this done?

The two SLC images overlap but do not match perfectly. For this reason it is necessary to resample one image to perfectly match with the other used as reference image

Preparation of data
Co-registration of SLC images
Computation of interferogram
Generation of differential interferogram
Computation of coherence
Phase unwrapping
Estimation of displacement

- In order to create an interferogram the SLC images must be precisely coregistered such that the residual offset is less than 1/10th or better.
- Generally a simple polynomial model is sufficient for the offset function between SLCs. In the case of large scale topography and a large baseline, this is no longer sufficient.



Images co-registration: procedure

Method 1

- Measure the offsets of the SLC images using the correlation of image intensities (just as done for the refinement of the lookup table in image geocoding) and use a model polynomial for resampling one image to the other

- Fast method but inaccurate in case of strong topography and diverging orbits

Method 2

- Establish a lookup table between the two SAR images using a DEM for correct correspondence between pixels.

- Compared to the offset method a finer co-registration should be achieved since topographic information is taken into account

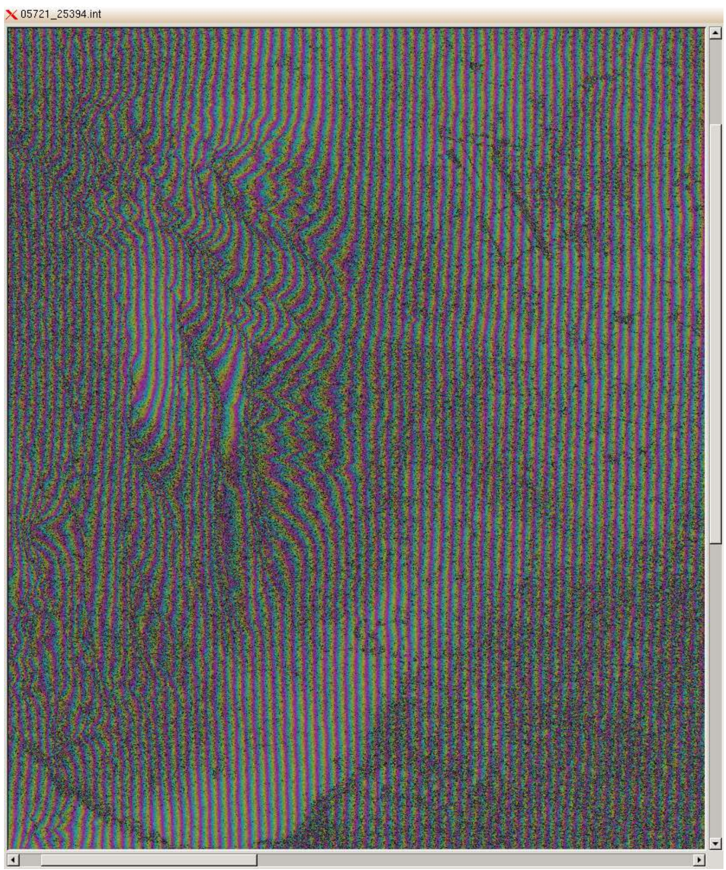
- Requirement: high-quality DEM

Computation of interferogram

What happens here?

At this point the complex interferogram is obtained from the co-registered SLCs, in addition the baseline is estimated.

- Preparation of data
- Co-registration of SLC images
- **Computation of interferogram**
- Generation of differential interferogram
- Computation of coherence
- Phase unwrapping
- Estimation of displacement



- The interferogram consists of
    - magnitude (correlation between images)
    - phase (InSAR phase)

The correlation indicates how accurate the phase information is (*fringe visibility*): the lower the correlation, the noisier the phase

The InSAR phase is a combination of several contributions (flat Earth, topography, atmospheric delays, noise etc.)

Baseline estimation

The interferometric baseline is defined as difference of platform position vectors when a given scatterer is imaged.

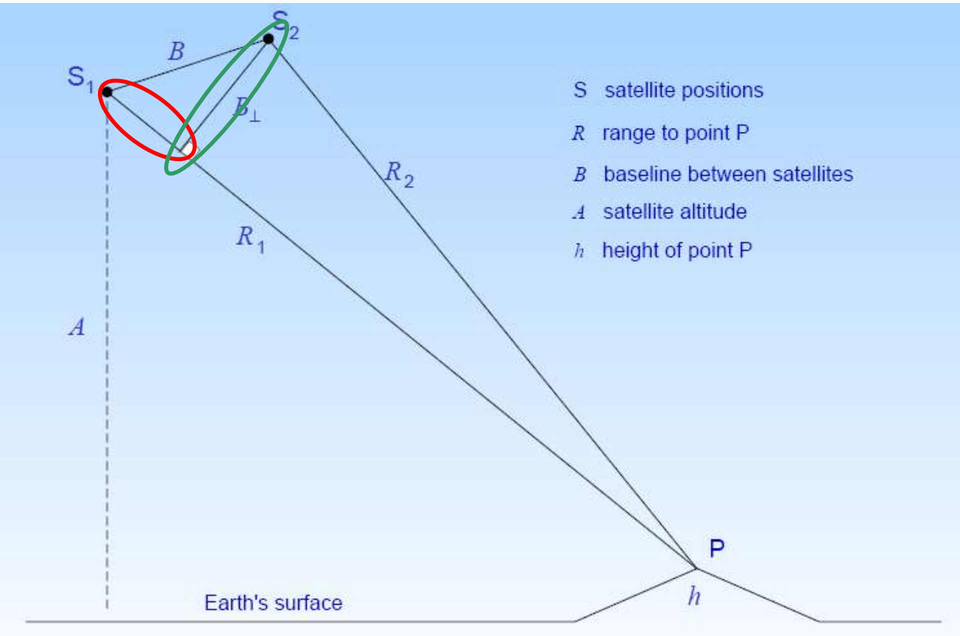
If the tracks are not parallel (typical for satellite orbits) the baseline changes along-track

The baseline can be decomposed into components. Typically the out of plane component is negligible so that we have 2 components only

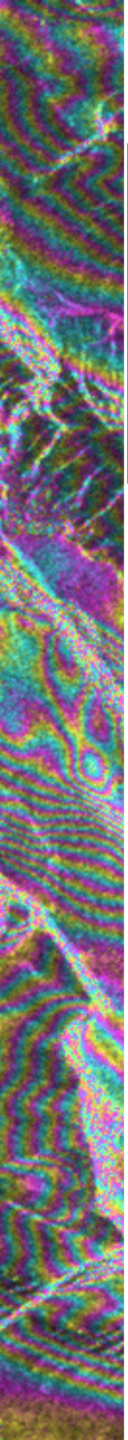
- The parallel baseline is the component along the line of sight
- The perpendicular baseline is the component perpendicular to the line of sight

The baseline can be estimated using

- Orbital information
- Fringe rate of the interferogram
- Ground control points (after unwrapping)



Source, GlobeSAR Program, CCRS



Generation of differential interferogram

Why is this done?

To obtain the phase containing only displacement information, we need to remove the flat Earth phase component and the topograhic phase component

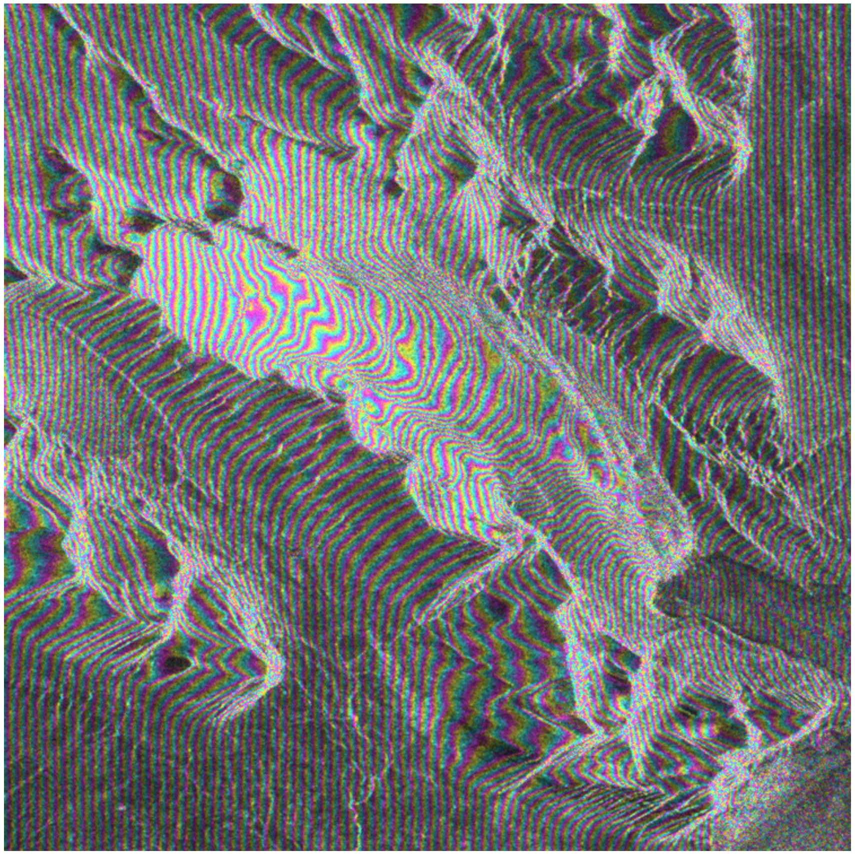
- Preparation of data
- Co-registration of SLC images
- Computation of interferogram
- Generation of differential interferogram
- Computation of coherence
- Phase unwrapping
- Estimation of displacement

The raw interferogram generated from the SLCs has
• linear phase trend across the image as a function of the slant range (*R*) (black box)
• modulation of phase due to topography (*z*); stronger modulation for longer baseline (B_n) (red box)

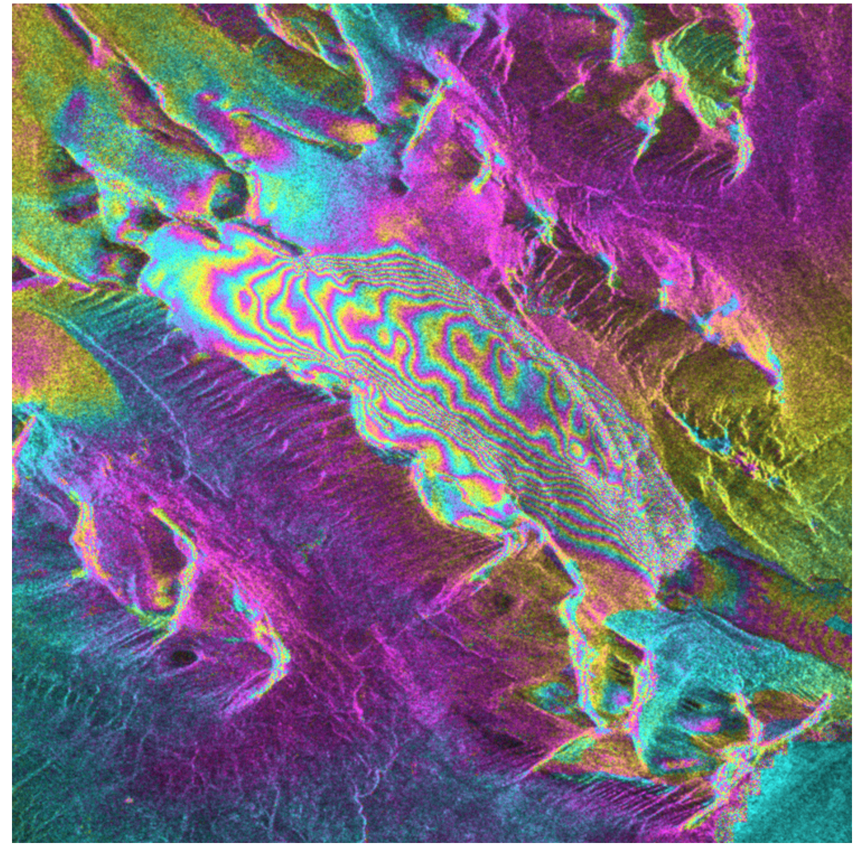
$$\phi = \boxed{-\frac{4\pi}{\lambda}R} + \boxed{\frac{4\pi B_n}{\lambda R \sin\theta} z} + \frac{4\pi}{\lambda}\eta + \frac{4\pi}{\lambda}\rho + \phi_{noise} + n \cdot 2\pi$$

These terms can be modelled and removed from interferogram, leaving fringes only related to displacement (as well as noise and atmospheric artifacts)

From original to differential interferogram



Raw interferogram



Differential interferogram

One fringe represents a variation of phase equal to 2π

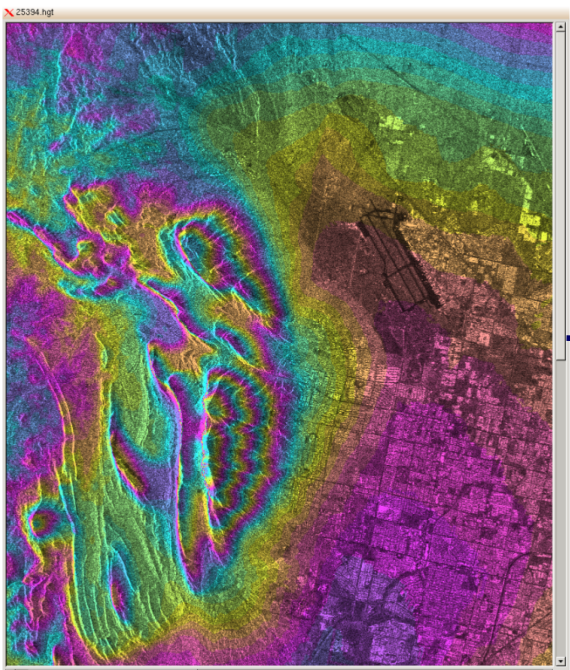
Motion (glacier) only in the center of the image

Generation of differential interferogram: Simulation of phase

Why is this done?

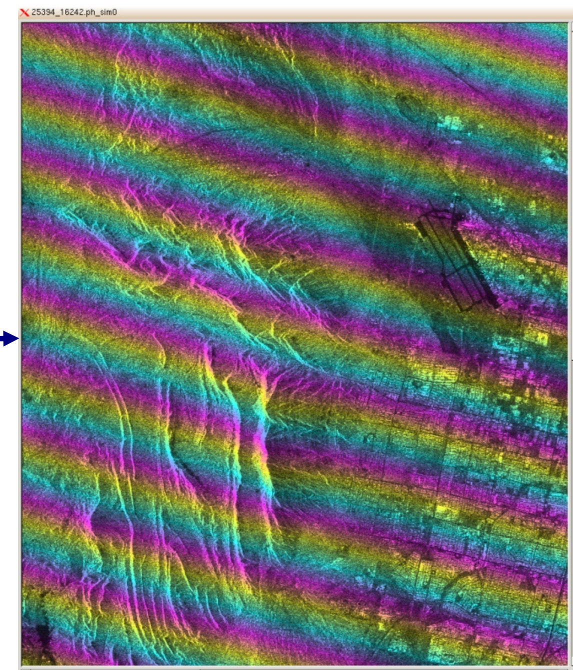
To obtain a differential interferogram, the phase due to flat Earth and topography has to be removed from the interferogram. The flat Earth fringes can be obtained from orbital information. The topographic phase can be generated from a DEM in SAR coordinates.

- Preparation of data
- Co-registration of SLC images
- Computation of interferogram
- Generation of differential interferogram
- Computation of coherence
- Phase unwrapping
- Estimation of displacement

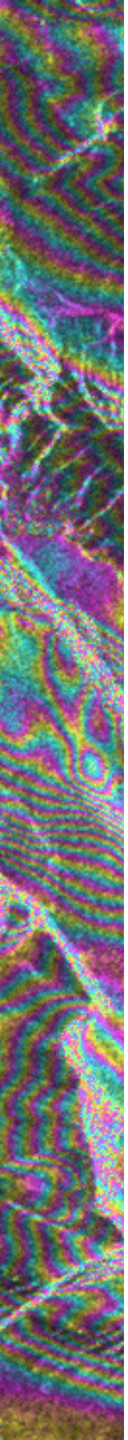


DEM in SAR geometry

Baseline estimate



Flat Earth + Topographic fringes



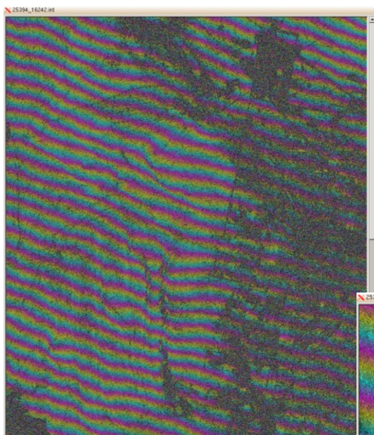
Generation of differential interferogram: Subtraction of phase

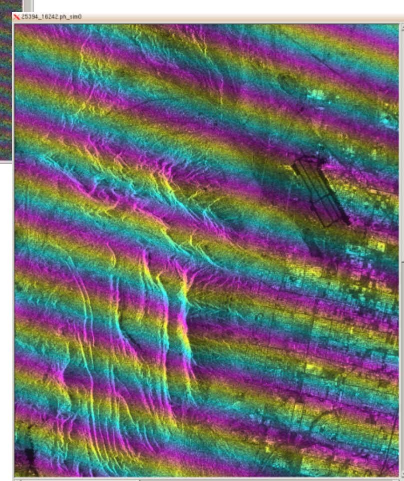
What happens here?

Subtracting the simulated phase from the original interferogram allows obtaining an (initial) differential interferogram, consisting of displacements, residual fringes due to inaccurate orbits and DEM errors, atmospheric artifacts, noise and the wrapping constant.

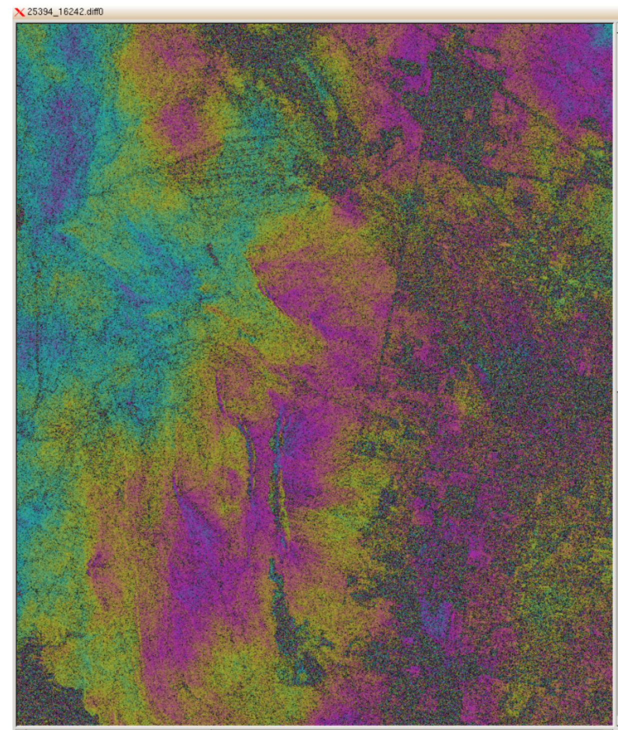
- Preparation of data
- Co-registration of SLC images
- Computation of interferogram
- Generation of differential interferogram
- Computation of coherence
- Phase unwrapping
- Estimation of displacement

Interferogram





Simulated fringes



Differential interferogram

Filtering of interferogram

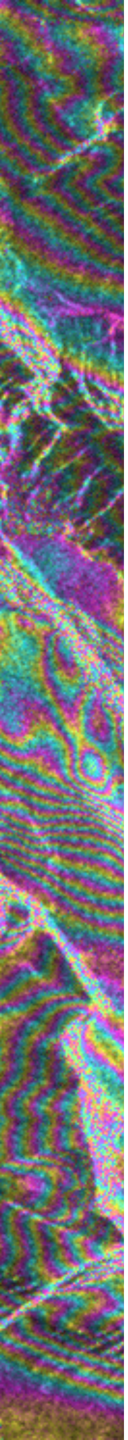
Why is this done?

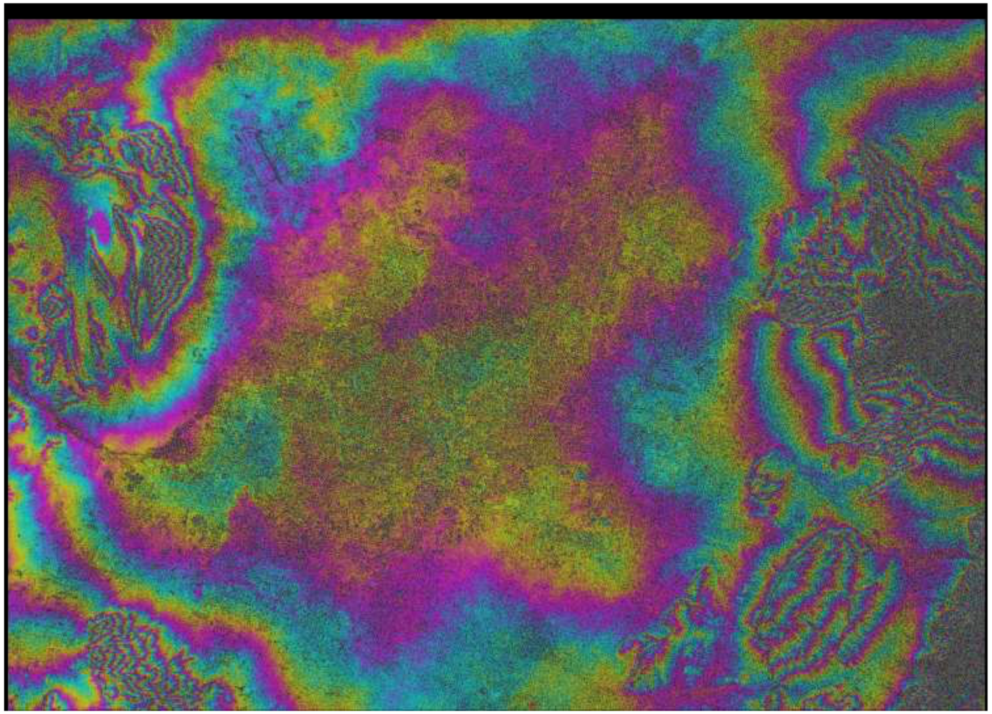
The interferometric phase can be affected by phase noise. To reduce it, multi-looking and filtering can be applied. In this way phase unwrapping will be simpler, more robust, more efficient.

Preparation of data
Co-registration of SLC images
Computation of interferogram
Generation of differential interferogram
Computation of coherence
Phase unwrapping
Estimation of displacement

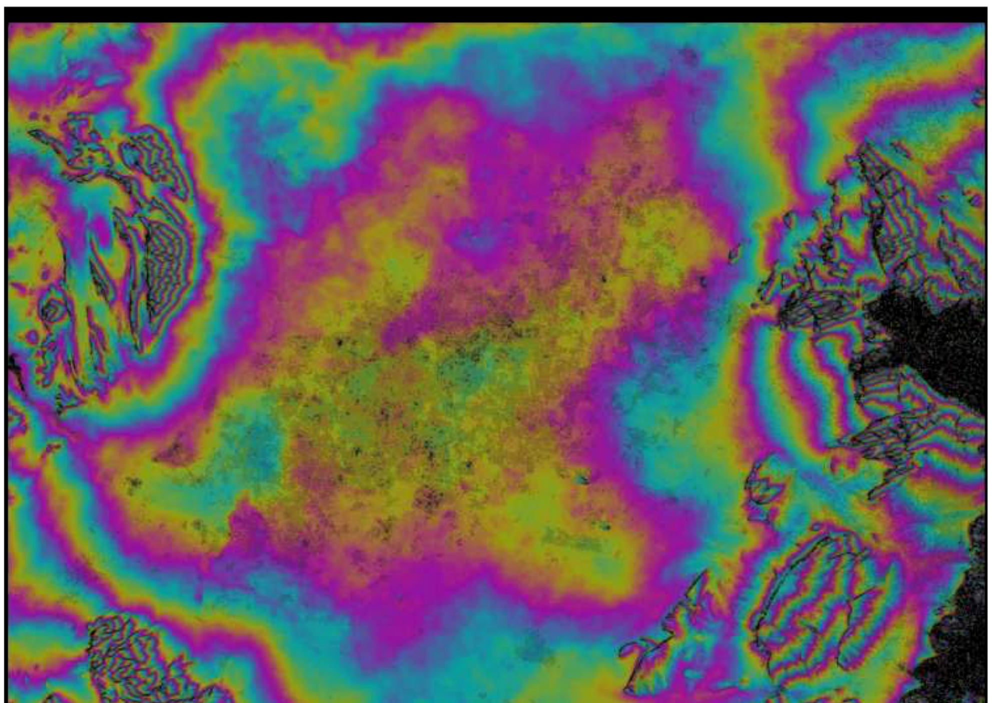
• There are several ways of filtering the InSAR phase

• Adaptive filtering with filtering function based on local fringe spectrum removes noise mantaining fringes (even dense ones)





Flattened phase, unfiltered



Flattened phase,
filtered with adaptive filtering

Coherence estimation

Why is this done?

Coherence is a measure of fringe visibility (=quality factor of phase) and can be used for thematic mapping of the area

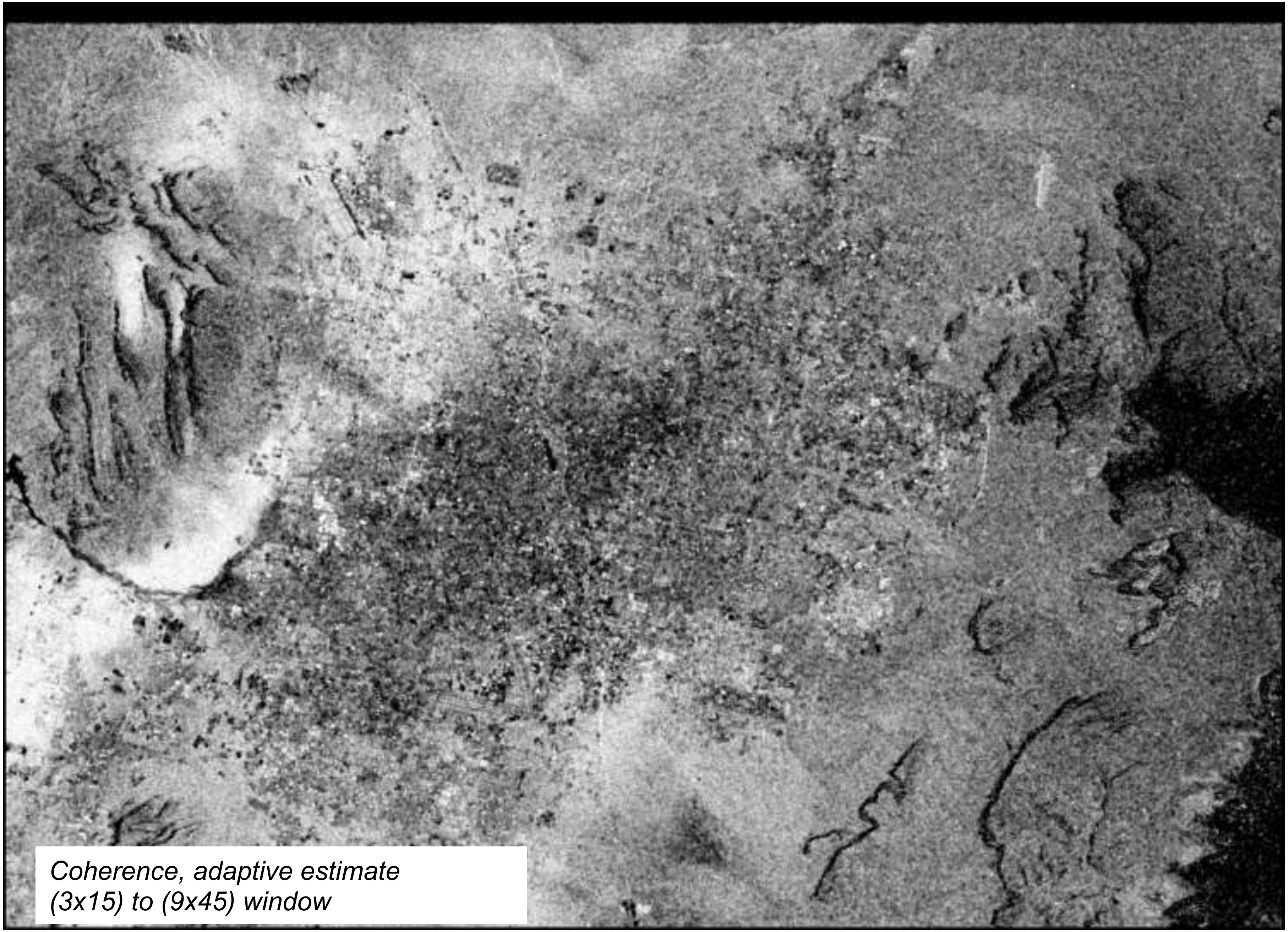
Preparation of data
Co-registration of SLC images
Computation of interferogram
Generation of differential interferogram
Computation of coherence
Phase unwrapping
Estimation of displacement

- Coherence is computed using a window of finite size.
- Coherence estimates have a bias and are more uncertain in regions of low coherence
- The larger the window the more accurate the coherence
- One method to limit the estimation errors is to use an adaptive estimation:

  use a window size and compute a first coherence image

  refine the coherence by using smaller/larger windows in high/low coherence areas


Coherence, 3x15 window



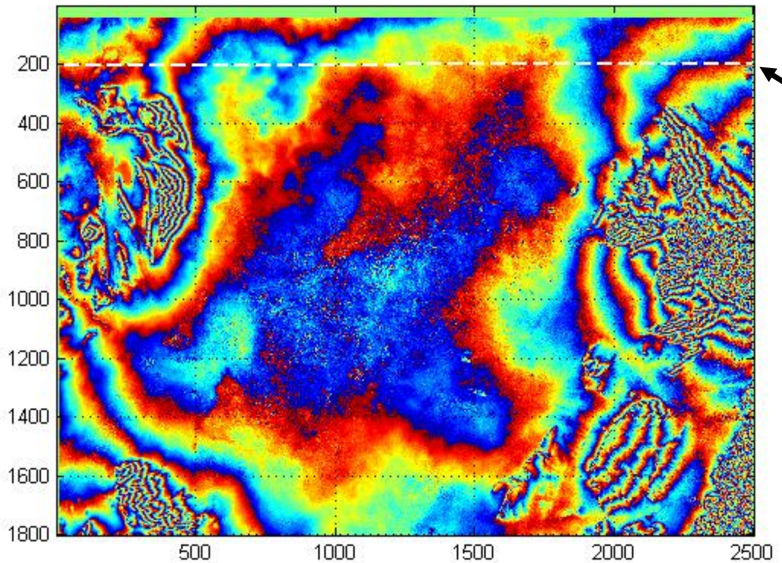
Coherence, adaptive estimate
(3x15) to (9x45) window

Phase unwrapping

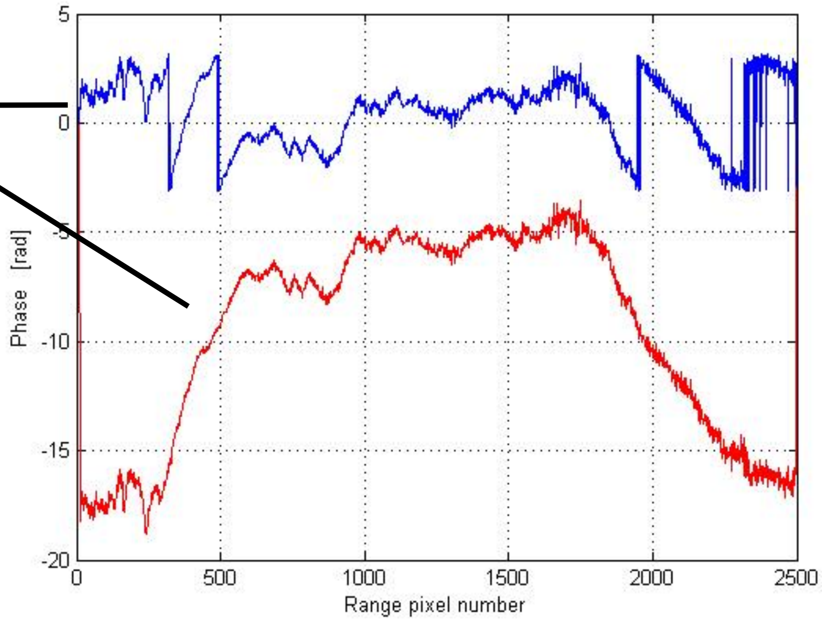
Why is this done?

At this stage, the phase of an interferogram has values in a 2π interval (e.g., $-\pi$ and $+\pi$). Phase unwrapping resolves the phase ambiguity and allows using the interferometric phase for estimation of displacement.

- Preparation of data
- Co-registration of SLC images
- Computation of interferogram
- Generation of differential interferogram
- Computation of coherence
- Phase unwrapping
- Estimation of displacement



Flattened Interferogram



Wrapped and unwrapped phase along transect

Phase unwrapping: approach

Phase unwrapping is based upon the assumption of smooth phase such that the phase differences are $|\Delta\phi| < \pi$ between adjacent samples.

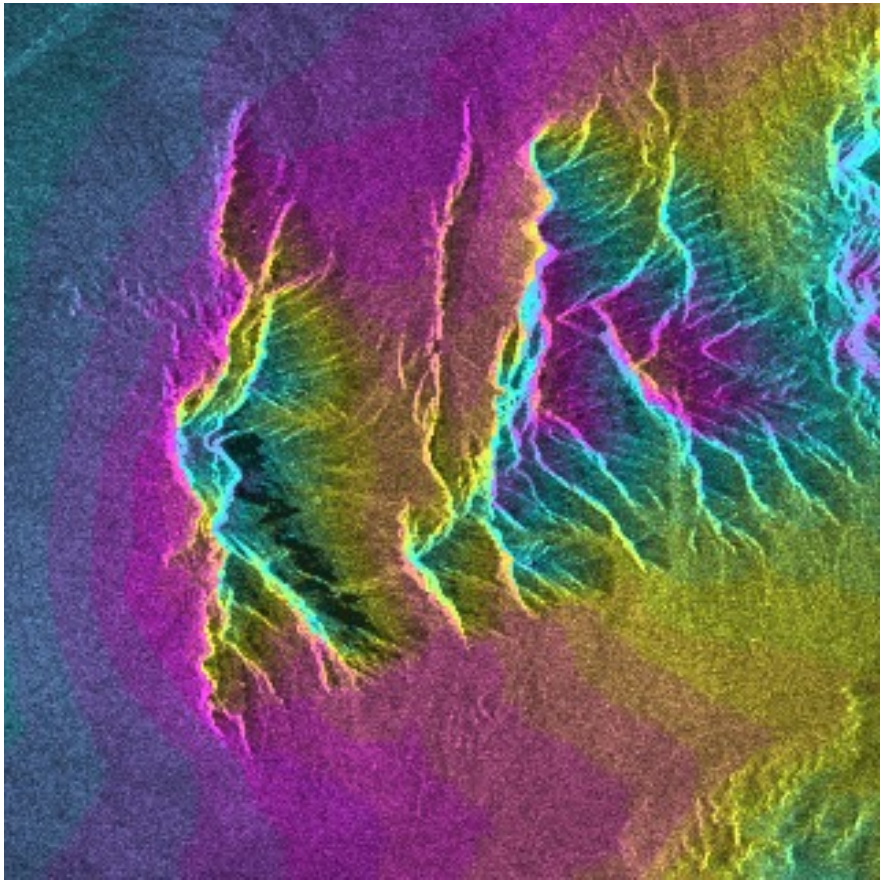
Unwrapping is done by integration of the phase differences using a region growing algorithm. The result must be independent of the integration path.

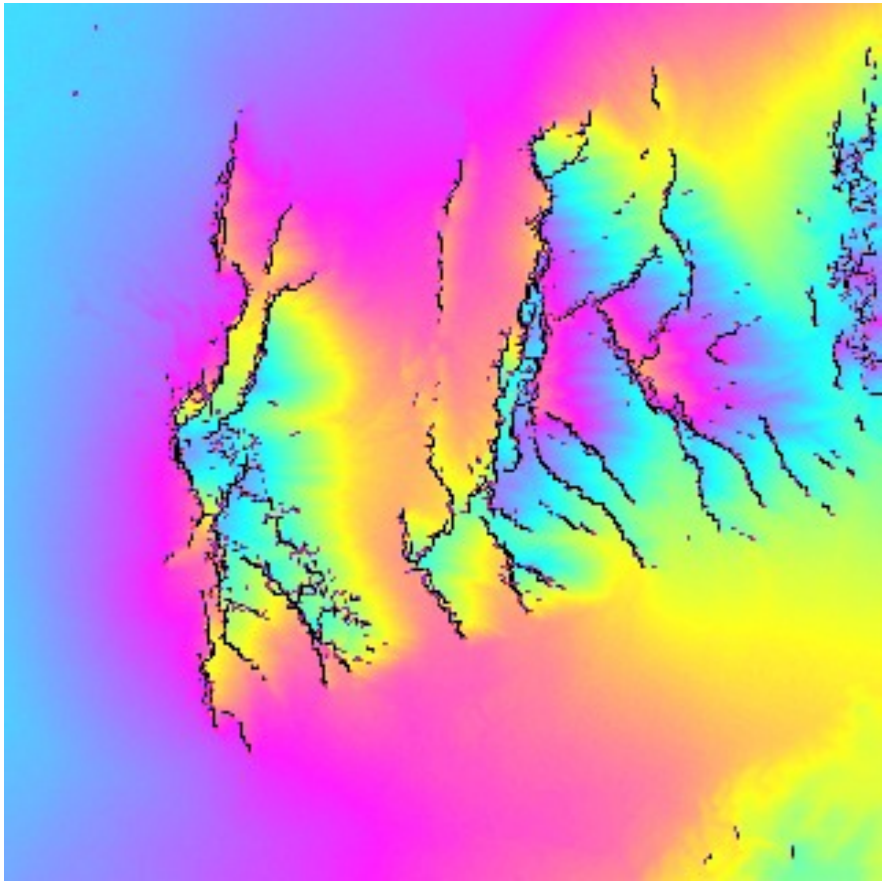
Rewrapping of the unwrapped phase should give the original phase values.

However $|\Delta\phi| > \pi$ occurs in interferograms for a number of reasons:

- <u>Phase noise</u>: Due to temporal decorrelation, baseline decorrelation, different Doppler centroids, shadow, or low SNR.
- <u>Phase undersampling</u>: Steep slopes can cause phase gradients to exceed π.
- <u>Phase discontinuities</u>: layover and discontinuous surface deformation (earthquake faults, glaciers) lead to phase jumps of multiples of 2π.

Example of phase unwrapping





Wrapped phase
(between fringes of the same color the phase difference is always 2π)

Unwrapped phase
(between fringes of the same color the phase difference is approximately 6π to highlight possible phase jumps - black areas are related to layover and have not been unwraapped)

Conversion of phase to displacement

What happens here?

After unwrapping, the displacement along the line of sight (LOS) can be simply obtained from the phase of the differential interferogram

Preparation of data
Co-registration of SLC images
Computation of interferogram
Generation of differential interferogram
Computation of coherence
Phase unwrapping
Estimation of displacement

$$\varphi = \frac{4\pi}{\lambda}\rho + \frac{4\pi}{\lambda}\eta$$

The differential phase φ is related to

- displacement along the line of sight, ρ
- as well as atmospheric delays, η

If atmospheric component is weak, the displacement is proportional to phase by a factor related to the wavelength: $\lambda/4\pi$

Overview

• Interferometric processing

• Introduction to the Interferometric SAR Processor (ISP) and Differential InSAR (DIFF&GEO) modules

•Use of ISP and DIFF&GEO for processing

• Summary

What is ISP? What can ISP do?

The Interferometric SAR Processor (ISP) encompasses a full range of algorithms required for generation of

- interferograms,
- height maps,
- coherence maps, and
- differential interferometric products

With the ISP the following steps, typical of InSAR processing, are performed

- pre-processing, generation of parameter files, calibration files and orbit data,
- precision registration of interferometric image pairs,
- interferogram generation (including common spectral band filtering),
- estimation of interferometric baseline,
- removal of flat Earth phase trend,
- estimation of coherence,
- adaptive filtering of interferograms,
- phase unwrapping
- precision estimation of interferometric baselines from ground control points,
- generation of topographic height, and
- image rectification and interpolation of interferometric height maps.

Processing related parameters and data characteristics are saved as text files that can be displayed using commercial plotting packages.

What is DIFF&GEO? What can DIFF&GEO do for DInSAR?

The Differential Interferometric & Geocoding software (DIFF&GEO) encompasses a full range of algorithms required for generation of

- differential interferograms for the 2-, 3- and 4-pass case,
- displacement maps from the differential interferogram
- stacking of interferograms

As well as programs for geocoding, i.e. for the transformation of images from the range-Doppler to any geographical coordinate system and vice versa.

With the DIFF the following steps, typical of DInSAR processing, are performed

- pre-processing, generation of parameter files,
- simulation of phases from height map
- subtraction of phases from one interferogram
- displacement computation
- combination of interferograms
- stacking of interferograms

Processing related parameters and data characteristics are saved as text files that can be displayed using commercial plotting packages.

DInSAR Processing with the GAMMA software

Let's see how to do differential SAR interferometry with the GAMMA software, which means let's see which programs are available for processing, what is their syntax and what are the main features

- pre-processing and interferometric processing
- simulation of interferometric phase using height map
- subtraction of phase (= generation of differential interferogram)
- phase unwrapping
- generation of displacement map

Overview

- Interferometric processing
- Introduction to the Interferometric SAR Processor (ISP) and Differential InSAR (DIFF&GEO) modules
- Use of ISP and DIFF&GEO for processing
- Summary

Dataset

Two images acquired by ALOS PALSAR over Christchurch, New Zealand, along the same track (RSP 337). SLC images have been generated with the MSP module.

ALOS PALSAR image 1: 20100813.slc (reference image)

ALOS PALSAR image 2: 20100928.slc ("slave" image)

For processing we need the ISP parameter files and the DEM in radar geometry

ALOS PALSAR image 1: 20100813.slc.par (generated with MSP module)

ALOS PALSAR image 2: 20100928.slc.par (generated with MSP module)

DEM in radar geometry: 20100813.hgt (generated with DIFF&GEO module)

Aim:

1) Co-register the two SLCs
2) Generate the interferogram
3) Compute the baseline
4) Simulate the flat Earth phase and the topographic phase from the DEM
5) Generate the differential interferogram & phase unwrapping
6) Compute the coherence
7) Estimate the displacement from the differential interferogram

Check of suitability of dataset for interferometry

Before proceeding with interferometric processing, we need to make sure the image pair is suitable. This means that the baseline must be significantly shorter than the critical baseline

The critical baseline in the case of ALOS PALSAR is about 4.5 km

The program requires the list of images for which the baseline shall be computed. The list (file SLC_tab) is generated with Linux command echo as follows (valid for a C-Shell)

echo "20100813.slc 20100813.slc.par" > SLC_tab

echo "20100928.slc 20100928.slc.par" >> SLC_tab

The baseline can be computed from orbital information included in the parameter files of the two SLC and a program called base_calc

```
base_calc SLC_tab 20100813.slc.par bperp.gr_file itab 1
```

The text file called bperp.gr_file contains the following information

1	20100813	20100928	-355.70410	45.9996	0.00000	45.99957	0.0000	-355.7041
	date1	*date2*	*baseline (m)*	*interval (days)*				

Comparison of SLC images

The two SLCs have different sizes and do not perfectly match.

From the SLC parameter files, we see:

20100813.slc.par	*20100928.slc.par*
Range samples: 5016	*Range samples: 5016*
Azimtuh lines: 27132	*Azimuth lines: 26721*

To display the two SLC, we use the program dis2SLC of the DISP module:

```
dis2SLC 20100813.slc 20100928.slc 5016 5016 15000 2000
```

(displaying a subset of 2000 lines, starting at line 15000)

Generation of multi-look intensities (MLI)

First of all, we generate the MLI images because they will be used throughout the processing.

To generate the MLIs, the program multi_look is used. Multi-look factors are 2 in range and 12 in azimuth.

multi_look 20100813.slc 20100813.slc.par 20100813.mli 20100813.mli.par 2 12

multi_look 20100928.slc 20100928.slc.par 20100928.mli 20100928.mli.par 2 12

These can be displayed with the program dis2pwr of the DISP module

dis2pwr 20100813.mli 20100928.mli 2508 2508

Co-registration

To make interferometry the two images need to match perfectly. For this:

1) The offsets between the two SLCs in range and azimuth need to be computed

2) The offsets vary across the image (orbits not perfectly parallel): this variation has to be taken into account

3) A polynomial has to be generated to model the offsets in range and azimuth all over the images

4) The slave image has to be resampled to match with the reference image using this polynomial

Images co-registration: procedure

1. Generation of a “co-registration” lookup table between multi-look images using height information

2. Resampling of the reference MLI to the geometry of the other MLI using this lookup table

2. Refinement of the lookup table by offset computation between resampled reference MLI and the other MLI

2. Resampling the second SLC to the reference SLC using the refined lookup table

2. Computation of residual offsets between reference SLC and resampled SLC

6. New resampling the second SLC to the reference SLC using the refined lookup table AND the estimate of the residual offsets

Images co-registration: step 1

1. Generation of a "co-registration" lookup table between multi-look images using height information

The lookup table represents the link between the geometries of the two images. It is generated based on orbital information and the DEM. Errors in the orbital information introduce errors in the lookup table.

The lookup table is generated with the program rdc_trans as follows

rdc_trans 20100813.mli.par 20100813.hgt 20100928.mli.par 20100813_20100928.lt

Images co-registration: step 2

2. Resampling of the reference MLI to the geometry of the other MLI using this lookup table

Resampling is done with the program geocode (see also GEO processing)

geocode 20100813_20100928.lt 20100813.mli 2508 20100928.sim 2508 2256 0 0

The reference MLI image (20100813.mli) has been resampled to the geometry of the other image. The result is saved to the file 20100928.sim

The two images, now in the same geometry, can be compared with the program dis2pwr of the DISP module

dis2pwr 20100928.sim 20100928.mli 2508 2508

The two images overlap nicely however a small mismatch is visible.

Images co-registration: step 3

3. Refinement of the lookup table by offset computation between resampled reference MLI and the other MLI

The lookup table is refined by

- estimating offsets between the two MLI images we have just compared
- using offsets to estimate coefficients of a model of the offsets
- applying the model to refine the lookup table

Images co-registration: step 3

- Generation of an offset parameter file (so called diff_par file, specific for MLI images) with program create_diff_par

This is an interactive program that requires the user to define values or accept default values

create_diff_par 20100928.mli.par - 20100813_20100928.diff_par 1 0

- Computation of local offsets between MLIs using few, large windows (12 x 12 windows, each 256 x 256 pixels large) using the program offset_pwrm

offset_pwrm 20100928.sim 20100928.mli 20100813_20100928.diff_par 20100813_20100928.offs 20100813_20100928.snr 256 256 20100813_20100928.offsets 2 12 12 7.0

- Estimation of offset model polynomial for MLIs (here just a constant) using the program offset_fitm

offset_fitm 20100813_20100928.offs 20100813_20100928.snr 20100813_20100928.diff_par 20100813_20100928.coffs 20100813_20100928.coffsets 0.2 1

Images co-registration: step 3

- New computation of local offsets with more and smaller windows (32 x 32 windows, each 128 x 128 pixels large), with program offset_pwrm

offset_pwrm 20100928.sim 20100928.mli 20100813_20100928.diff_par 20100813_20100928.offs 20100813_20100928.snr 128 128 20100813_20100928.offsets 2 32 32 0.2

- Estimation of offset polynomial model based on new set of offsets (now we use a linear function) with program offset_fitm

offset_fitm 20100813_20100928.offs 20100813_20100928.snr 20100813_20100928.diff_par 20100813_20100928.coffs 20100813_20100928.coffsets 0.2 3

Images co-registration: step 3

Result of offset estimation between MLIs

total number of culling iterations: 3
final solution: 735 offset estimates accepted out of 1024 samples

final range offset poly. coeff.: 0.09612 -7.65680e-06 -2.16305e-05
final azimuth offset poly. coeff.: -0.00503 2.16646e-05 -2.82372e-05

final range offset poly. coeff. errors: 3.99444e-04 1.55454e-07 2.67923e-07
final azimuth offset poly. coeff. errors: 2.37594e-04 9.24657e-08 1.59364e-07
final model fit std. dev. (samples) range: 0.1330 azimuth: 0.0791

- Refinement of lookup table with the model polynoimial using the program gc_map_fine

gc_map_fine 20100813_20100928.lt 2508 20100813_20100928.diff_par 20100813_20100928.lt_fine 1

Images co-registration: step 4

4. Resampling the second SLC to the reference SLC using the refined lookup table

Resampling using the refined lookup table is supported by the program SLC_interp_lt

SLC_interp_lt 20100928.slc 20100813.slc.par 20100928.slc.par 20100813_20100928.lt_fine 20100813.mli.par 20100928.mli.par - 20100928.rslc 20100928.rslc.par -

The program generates a resampled SLC image of the "slave" image (20100928.rslc) and the corresponding SLC parameter file.

We can compare the reference SLC and the resampled SLC with the program dis2SLC of the DISP module

dis2SLC 20100813.slc 20100928.rslc 5016 5016 15000 2000

(displaying a subset of 2000 lines, starting at line 15000)

Images co-registration: step 5

5. Computation of residual offsets between reference SLC and resampled SLC

Residual offsets are estimated by cross-correlating the SLC images (reference and resampled) using a similar procedure compared to what just used before.

NOTE: the procedure is similar, the programs are in part different

The procedure to compensate for residual offsets is the following:
- estimate offsets between the two SLC images (reference and resampled)
- use offsets to estimate coefficients of a model of the offsets
- apply the model to obtain a new image of the resampled SLC image

Images co-registration: step 5

- Generation of an offset parameter file (so called off_par file, specific for SLC data) with program create_offset

This is an interactive program that requires the user to define values or accept default values (in this case, just press "enter" each time)

create_offset 20100813.slc.par 20100928.rslc.par 20100813_20100928.off 1 - - 0

- Computation of local offsets between SLCs using few, large windows (12 x 12 windows, each 256 x 256 pixels large) using the program offset_pwr

offset_pwr 20100813.slc 20100928.rslc 20100813.slc.par 20100928.rslc.par 20100813_20100928.off 20100813_20100928.offs 20100813_20100928.snr 256 256 20100813_20100928.offsets 1 12 12 7.0

- Estimation of offset model polynomial for SLC offsets (here a linear function) using the program offset_fit

offset_fit 20100813_20100928.offs 20100813_20100928.snr 20100813_20100928.off 20100813_20100928.coffs 20100813_20100928.coffsets 0.2 3

Generation of ISP offset parameter file

NOTE: It is possible to make the generation of the offset parameter file automatic by saving the input values to a dummy text file from which the program create_offset reads the required parameter values

In this example the dummy file is called off_in

```
# Writing parameter values to the dummy text file off_in
echo “interferogram parameters” > off_in
echo “0  0” >> off_in
echo “32  32” >> off_in
echo “64  64” >> off_in
echo “7.000” >> off_in
echo “0” >> off_in
echo “5016” >> off_in
```

```
# Generating offset parameter file by reading required parameters from the off_in file
create_offset 20100813.slc.par 20100928.rslc.par 20100813_20100928.off 1 1 < off_in
```

NOTE: the same can be applied to generate the diff_par file with create_diff_par program. The parameters should be adapted based on the values requested by this program.

Images co-registration: step 5

- New computation of local offsets with more and smaller windows (32 x 32 windows, each 128 x 128 pixels large), with program offset_pwr

offset_pwr 20100813.slc 20100928.rslc 20100813.slc.par 20100928.rslc.par 20100813_20100928.off 20100813_20100928.offs 20100813_20100928.snr 128 128 20100813_20100928.offsets 2 32 32 0.2

- Estimation of offset polynomial model based on new set of offsets (again a linear function) with program offset_fit

offset_fit 20100813_20100928.offs 20100813_20100928.snr 20100813_20100928.off 20100813_20100928.coffs 20100813_20100928.coffsets 0.2 3

Images co-registration: step 5

Result of offset estimation between SLCs

total number of culling iterations: 5
final solution: 442 offset estimates accepted out of 1024 samples

final range offset poly. coeff.: -0.17178 7.81189e-06 4.85910e-06
final range offset poly. coeff. errors: 2.96976e-04 6.20941e-08 1.65762e-08

final azimuth offset poly. coeff.: 0.21825 -1.00156e-04 1.42818e-05
final azimuth offset poly. coeff. errors: 7.49105e-04 1.56629e-07 4.18126e-08

final model fit std. dev. (samples) range: 0.0300 azimuth: 0.0758

- Estimation based on large number of offsets (> 50% of initial samples, red values)
- Small offsets estimated (green values) and small estimation errors (blue values)
- Co-registration error less than 1/10th pixel size in range (orange values)

Images co-registration: step 6

6. Resampling the second SLC to the reference SLC using the refined lookup table AND residual offsets estimate

Resampling using the refined lookup table AND residual offsets is supported by the program SLC_interp_lt

SLC_interp_lt 20100928.slc 20100813.slc.par 20100928.slc.par 20100813_20100928.lt_fine 20100813.mli.par 20100928.mli.par 20100813_20100928.off 20100928.rslc 20100928.rslc.par -

The program generates a new resampled SLC image of the "slave" image (20100928.rslc) and the corresponding SLC parameter file.

We can compare the reference SLC and the resampled SLC with the program dis2SLC of the DISP module

dis2SLC 20100813.slc 20100928.rslc 5016 5016 15000 2000

(displaying a subset of 2000 lines, starting at line 15000)

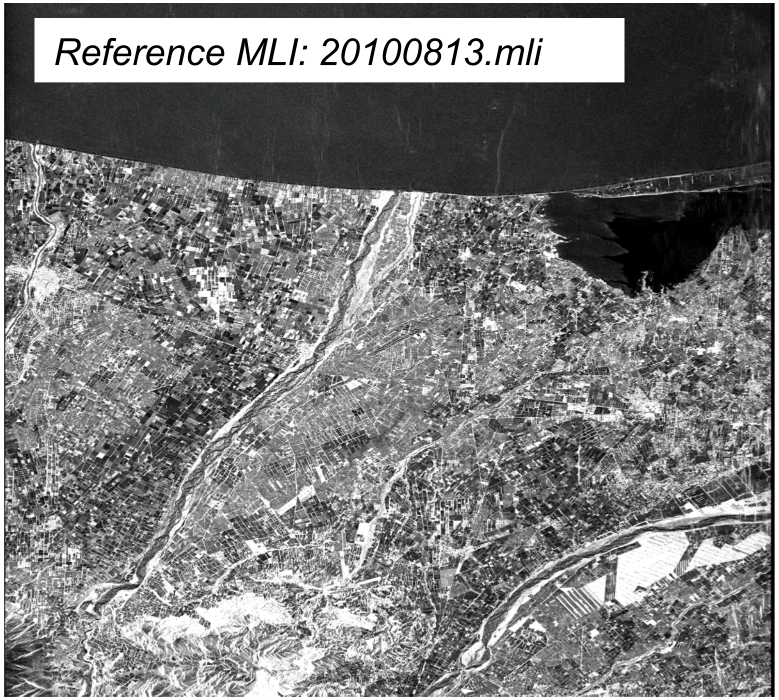
Generation of MLI image of resampled SLC

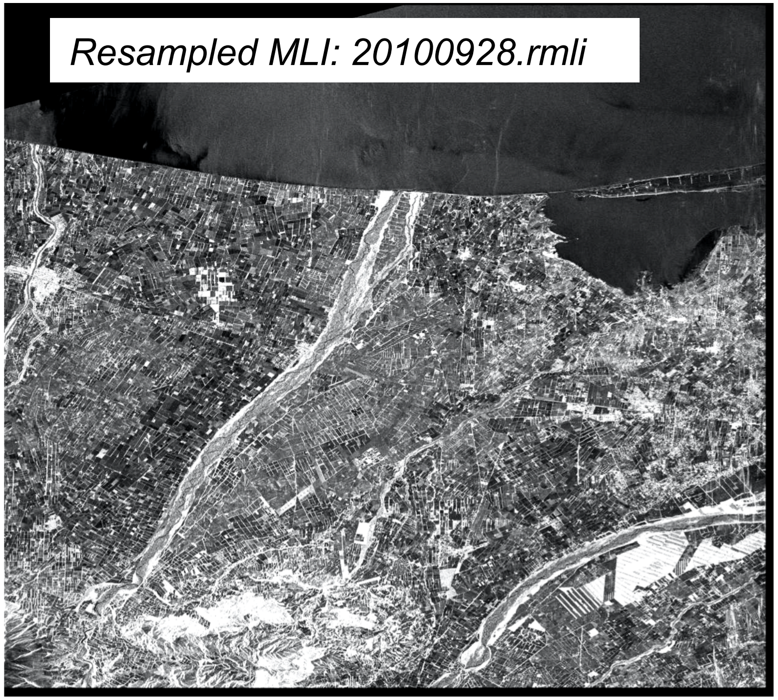
In the future, we will also need the MLI image of the resampled SLC, which can be generated with the program multi_look

multi_look 20100928.rslc 20100928.rslc.par 20100928.rmli 20100928.rmli.par 2 12

The reference MLI image and the resampled MLI image can be compared with the program dis2pwr of the DISP module

dis2pwr 20100813.mli 20100928.rmli 2508 2508





Computation of raw interferogram

This step is actually not necessary in our processing because the GAMMA Software has a program to compute the differential interferogram directly from the two SLC images.

Generation of the raw interferogram is however shown for educational purposes.

Computation of a raw interferogram using the reference SLC and the resampled slave SLC is done with the program SLC_intf. To compute the interferogram, we use multi-look factors 2 in range and 12 in azimuth. In this way some noise will be removed and the interferogram will appear more "comprehensible" compared to a full resolution interferogram.

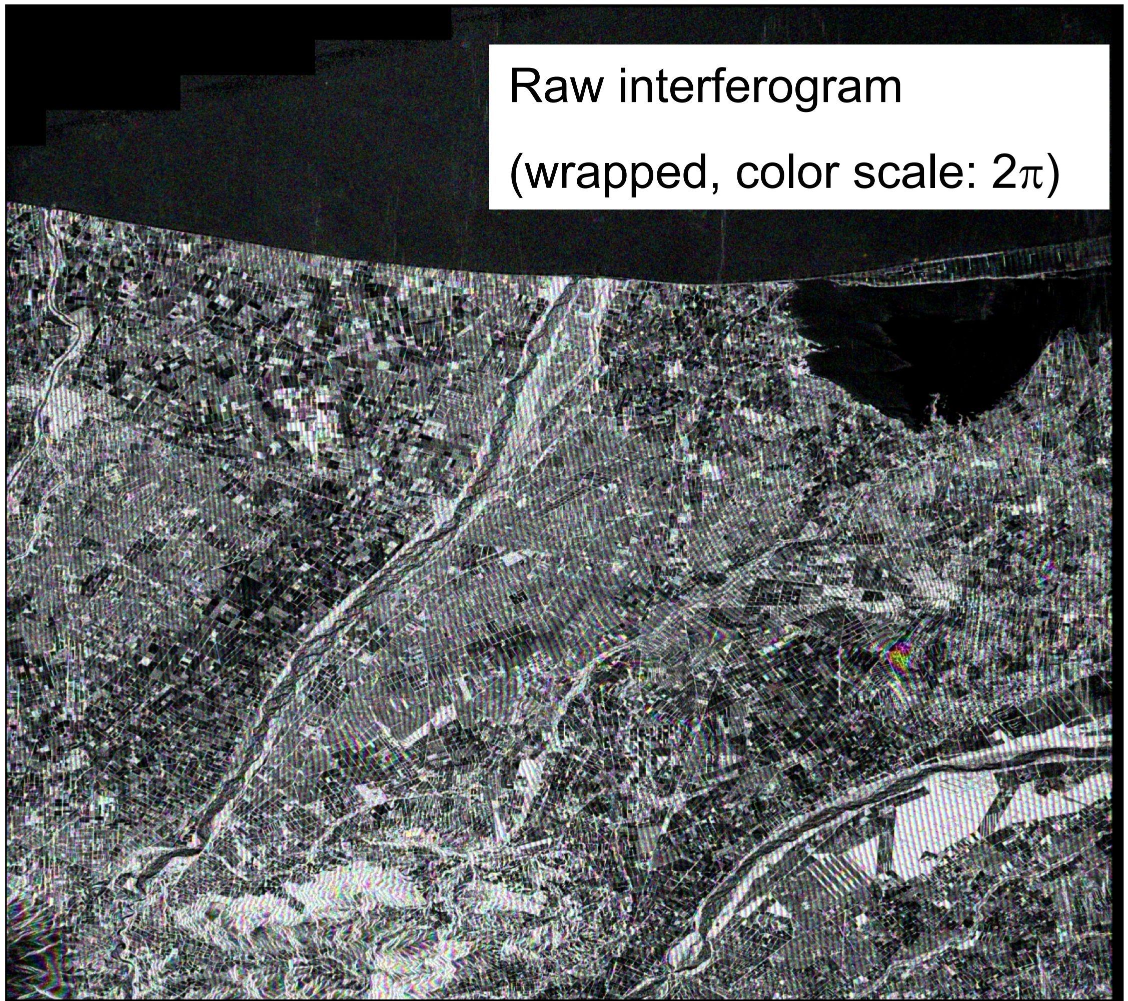
```
SLC_intf 20100813.slc 20100928.rslc 20100813.slc.par 20100928.rslc.par 20100813_20100928.off 20100813_20100928.int 2 12
```

- Input are the two overlapping SLCs and the parameter files, and the *.off file
- Output is the interferogram (*.int) file
- The offset parameter file is updated in case multilook factors different than 1 are used

Display of interferogram

We can display the interferogram (fringes) with the intensity information in the background with the program dismph_pwr of the DISP module

dismph_pwr 20100813_20100928.int 20100813.mli 2508



Baseline estimation

To determine the differential interferogram, an estimate of the baseline is needed.

There are different methods for obtaining this estimate. One is based on orbital information. For PALSAR data, this method is reliable since orbital information delivered by JAXA is rather accurate.

The program supporting the estimation of the baseline from orbital data is called base_orbit

base_orbit 20100813.slc.par 20100928.rslc.par 20100813_20100928.base

Orbital data is read from the SLC parameter files of the two images and the baseline is written to a text file.

The baseline estimate (for the center of the image) is shown on the screen

baseline parallel component (m): -231.3531
baseline perpendicular component (m): -355.7041

Simulation of interferometric phase

To determine the differential interferogram, an estimate of the phase due to flat Earth and topographic components is also needed.

To simulate an interferometric phase using a height map we use the program phase_sim_orb

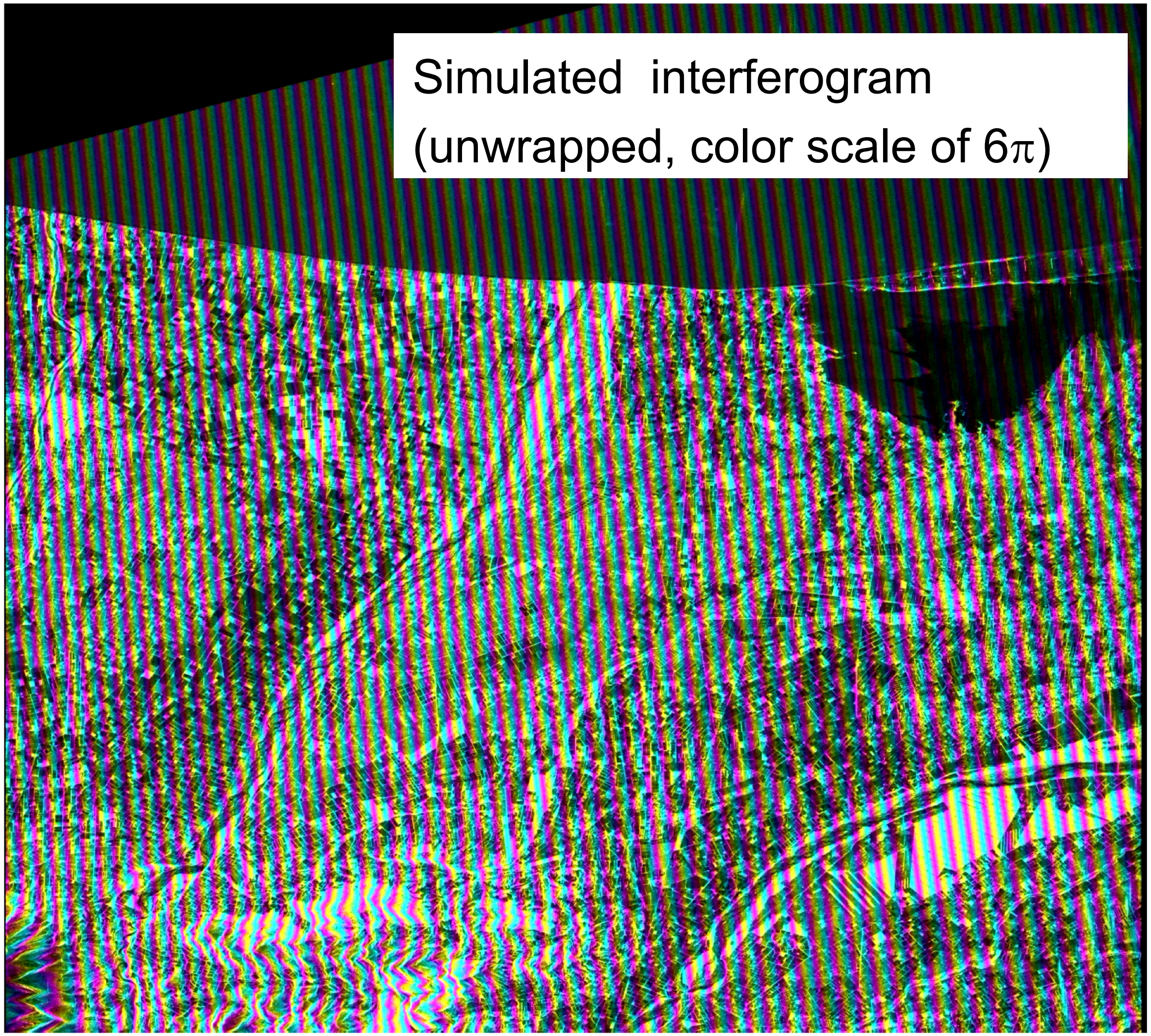
```
phase_sim_orb 20100813.slc.par 20100928.rslc.par 20100813_20100928.off 20100813.hgt 20100813_20100928.phase_sim
```

The program computes a simulated, unflattened and unwrapped interferogram (*.phase_sim) from the height map using the baseline estimate from the orbits.

Display of simulated interferometric phase

To display the unwrapped simulated phase we can use the DISP program specific for unwrapped phase images: disrmg

disrmg 20100813_20100928.phase_sim 20100813.mli 2508



Generation of differential interferogram

The generation of the differential interferogram from the SLC images and the simulated phase is supported by the program SLC_diff_intf

SLC_diff_intf 20100813.slc 20100928.rslc 20100813.slc.par 20100928.rslc.par 20100813_20100928.off 20100813_20100928.phase_sim 20100813_20100928.diff0 2 12 1 1 0.25 1 1

Input

- the 2 SLC images (reference and resampled)
- the simulated, unwrapped interferogram (*.phase_sim)
- the interferogram parameter file (*.off)

Output

- the differential interferogram, (*.diff0)

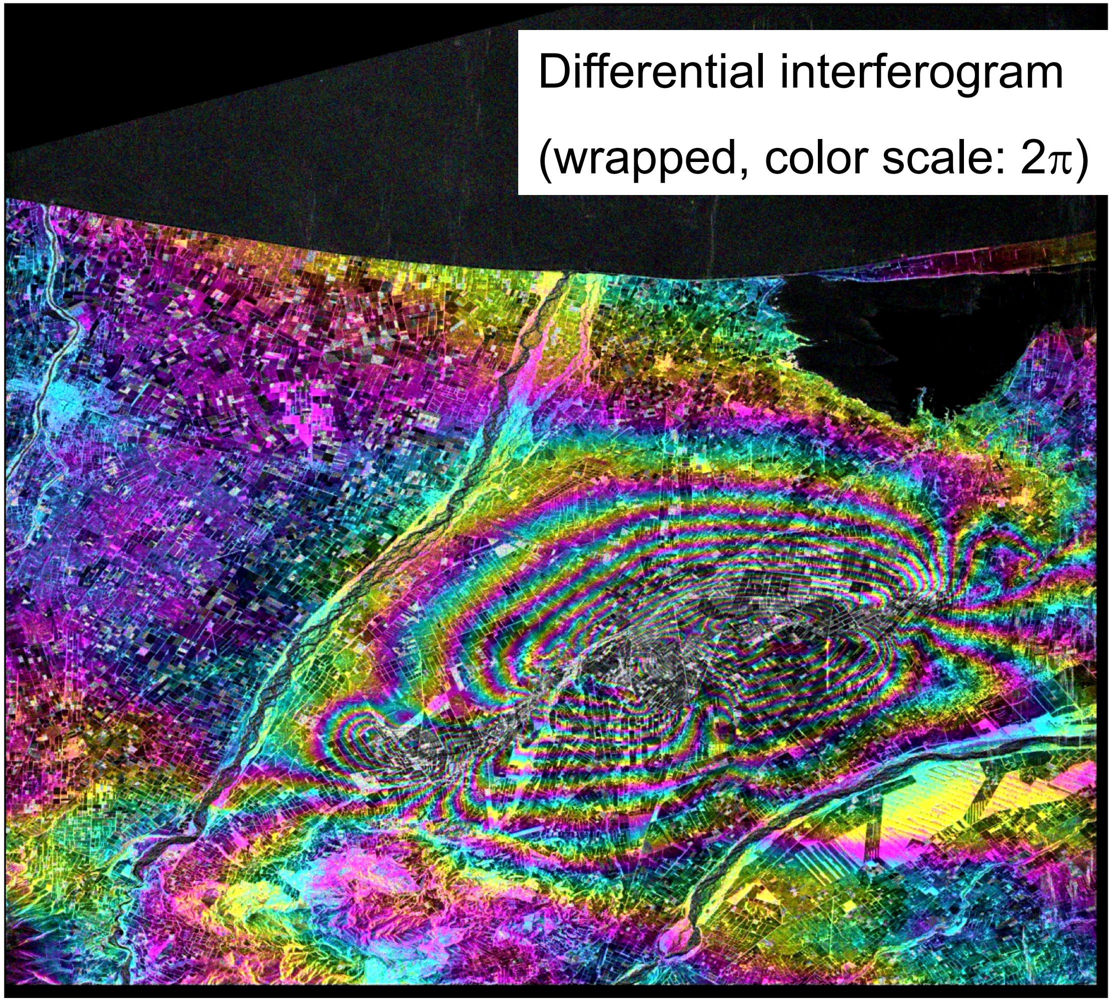
The program offers the possibility to choose

- multi-look factors to reduce noise in the interferogram (here 2 in range and 12 in azimuth)
- some filtering option

Display of interferogram

We can display the differential interferogram (fringes) with the intensity information in the background with the program dismph_pwr of the DISP module

```
dismph_pwr 20100813_20100928.diff0 20100813.mli 2508
```



Coherence estimation with the ISP

Having available a differential interferogram, we can compute the coherence image.

The ISP offers the program cc_wave [LAT offers the advanced program cc_ad]

- cc_wave uses a fixed window size
- cc_ad uses an adaptive window size

Input are the interferogram and the 2 intensity images

Output is the coherence image

The user can choose the window size for the estimation of coherence

Windowing can be used (= less weight to external pixels) to avoid too much loss of resolution when using large windows (e.g. in areas of low coherence)

Coherence estimation

Let's see what cc_wave produces

cc_wave 20100813_20100928.diff0 20100813.mli 20100928.rmli 20100813_20100928.cc 2508 5 5 1

- Inputs are the interferogram (*.diff0) and the 2 MLI images
- Output is the coherence image (*.cc)
- Window size used is 5 x 5 (notice: we used before a 2x12 multi-look so it is as if we were computing coherence using a 10x60 window)
- Triangular windowing is used (= less weight to external pixels) to avoid too much loss of resolution

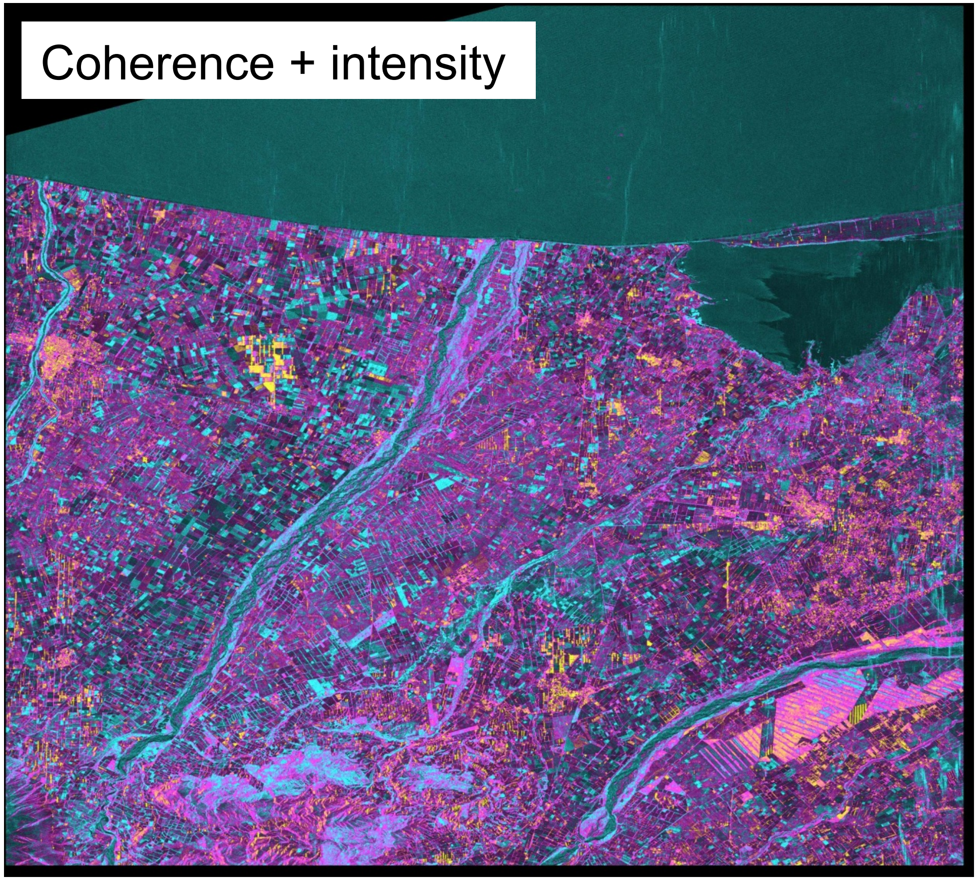
Display of coherence image

We can display the coherence image with the intensity information in the background with the program discc of the DISP module

discc 20100813_20100928.cc 20100813.mli 2508

Coherence image only with dis_linear:

dis_linear 20100813_20100928.cc 2508



Adaptive filtering of interferogram

Any interferogram can be filtered to remove noise.

The software offers several filters. An advanced filter in the ISP is implemented in the program adf

- The filter computes locally the interferogram power spectrum, designs a filter based on the power spectrum, filters the interferogram, estimates the phase noise coherence value for the filtered interferogram and writes out the filtered interferogram and coherence map
- The goal of the adaptive filtering step is to reduce phase noise thereby reducing the number of residues
- Filtering of the interferogram reduces the noise and hence the total number of residues. Too strong filtering may introduce phase unwrapping errors by eliminating fringes
- Multiple application of the filter is possible, and can lead to better results when using large values of the filter exponent parameter

Adaptive filtering of interferogram

To filter the inteferogram in our example we use the program adf as follows

```
adf  20100813_20100928.diff0  20100813_20100928.diff0_sm  20100813_20100928.diff0_smcc
2508 0.5 32 7 8 0 0 0.25
```

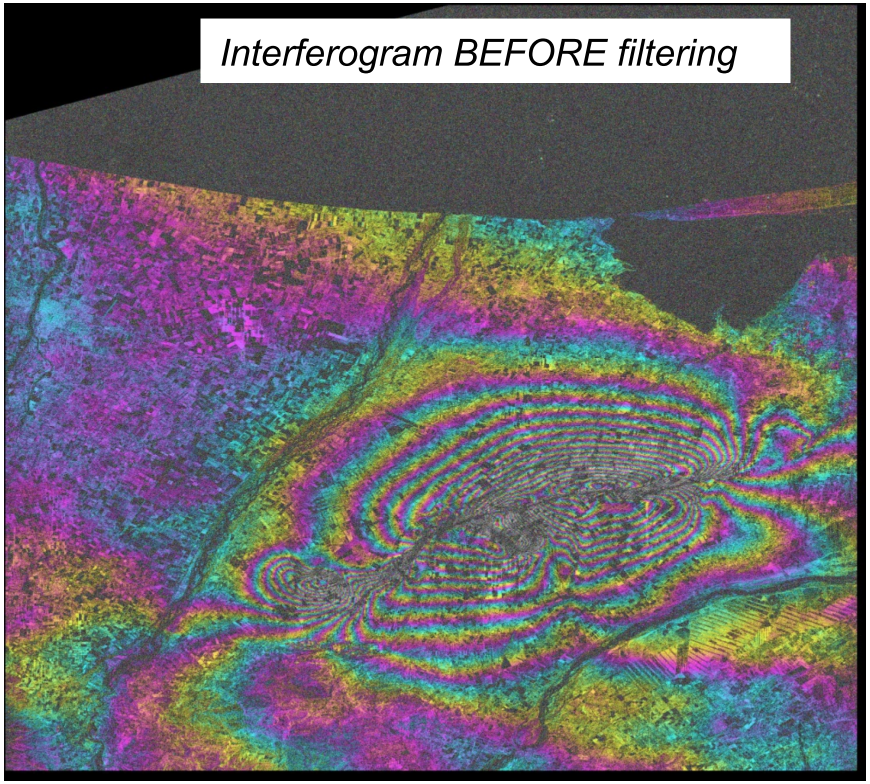
The program generates:

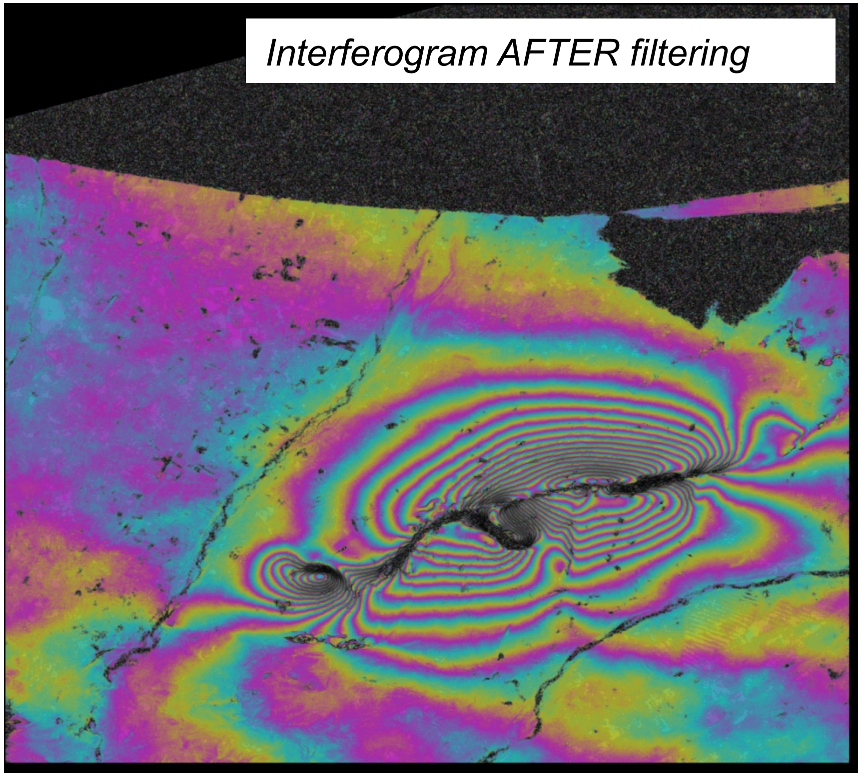
- filtered differential interferogram: *.diff0_sm
- filtered correlation of interferogram: *.diff0_smcc

Display of differential interferogram before and after adf

The differential interferogram before and after filtering can be displayed with the program dis2mph of the DISP module

dis2mph 20100813_20100928.diff0 20100813_20100928.diff0_sm 2508 2508





Phase unwrapping

The GAMMA software provides programs for two algorithms of phase unwrapping. Here we use the program supporting the Minimum Cost Flow (MCF) algorithm. For phase unwrapping, we therefore use the program mcf

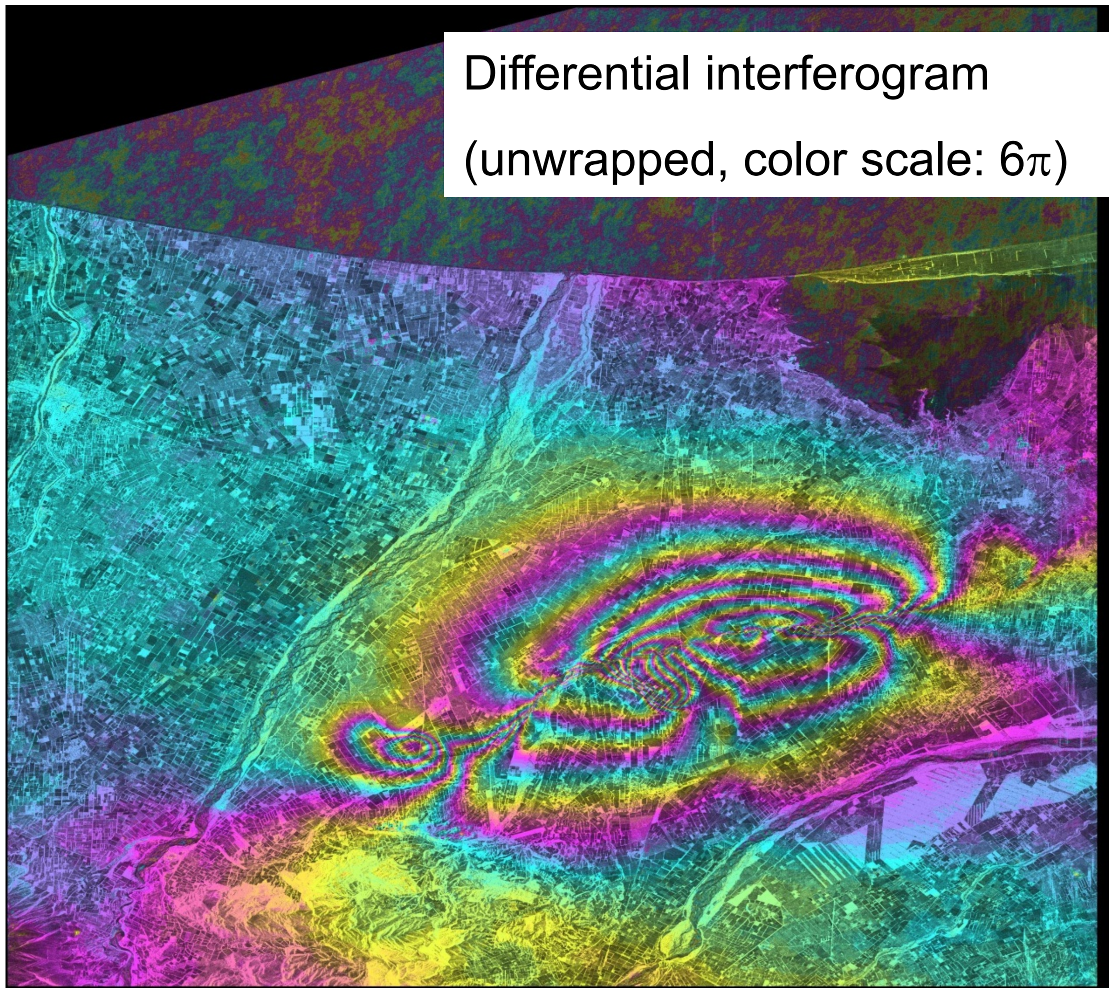
mcf 20100813_20100928.diff0_sm 20100813_20100928.cc - 20100813_20100928.diff0_sm.unw 2508 0 0 0 - - 1 1 512 200 750 1

- Input are the flattened interferogram and the coherence image
- The phase is unwrapped by considering the lowest costs, which are determined by the coherence image
- Output is the unwrapped phase
- The program has several parameters one can play with. Here, we
  - used a Delaunay Triangulation,
  - considered the whole image
  - used 1 patch (use more if data file is big -> causing memory allocation problems)
  - used as starting point for unwrapping the position 200,750 i.e. a stable point with high coherence and clear phase.

Display of unwrapped interferogram

We can display the unwrapped differential interferogram with the intensity information in the background with the program disrmg of the DISP module.

disrmg 20100813_20100928.diff0_sm.unw 20100813.mli 2508



Issues related to phase unwrapping

The unwrapped interferogram presents a discontinuity at the top right part of the image, nearby the lagoon. This means that something went wrong.

The problem is that this area is surrounded by an extended region with low coherence (water) and is connected with high coherence area with a few pixels only. The MCF algorithm is fooled and is not able to find the correct compensation for the residues.

The solution to the problem could be to test

- Use a different image for the weight (smoothed correlation instead of coherence)
- Unwrapping the original interferogram and not the filtered version
- Mask out the trouble area and unwrap separately. Then merge results.
- Unwrapping a multi-looked version of the original interferogram (less noisy) and then oversample back to the original pixel size

Computation of displacements

To determine the displacement from the unwrapped differential interferogram we use the program dispmap

dispmap 20100813_20100928.diff0_sm.unw 20100813.hgt 20100813.slc.par 20100813_20100928.off 20100813_20100928.disp 0

Input

- the unwrapped differential interferogram (*.diff0_sm.unw)
- the height map (*.hgt)
- the parameter files of the reference image (*.slc.par) and of the interferogram (*.off)

Output

- the displacement map (*.disp)

The differential interferometric phase corresponds to the displacement along the SAR look vector. As a consequence the 3-dimensional displacement of a surface element cannot be completely described.

To let dispmap show in a clear way displacements the user can select the value of the “mode” flag on the command line.

Use of dispmap and interpretation of displacement map

Under the assumption of a predefined surface displacement direction a conversion to vertical or horizontal displacement is possible by setting the „mode“ flag as follows

0: conversion to displacement along the look vector in meters

+ signs correspond to displacement towards the sensor

For this conversion no specific assumption on the surface deformation is necessary.

1: conversion to vertical displacement in meters

+ signs correspond to increasing surface height

- signs correspond to subsidence

Vertical displacement can often be assumed in the case of subsidence

2: conversion to horizontal displacement component in ground range direction in meters

+ corresponds to decreasing ground range

Display of displacement map

To display the displacement map together with the intensity MLI in the background, we use the program disdt_pwr24 of the DISP module:

disdt_pwr24 20100813_20100928.disp 20100813.mli 2508 - - - 1

• A color cycle of 1 meter is used.

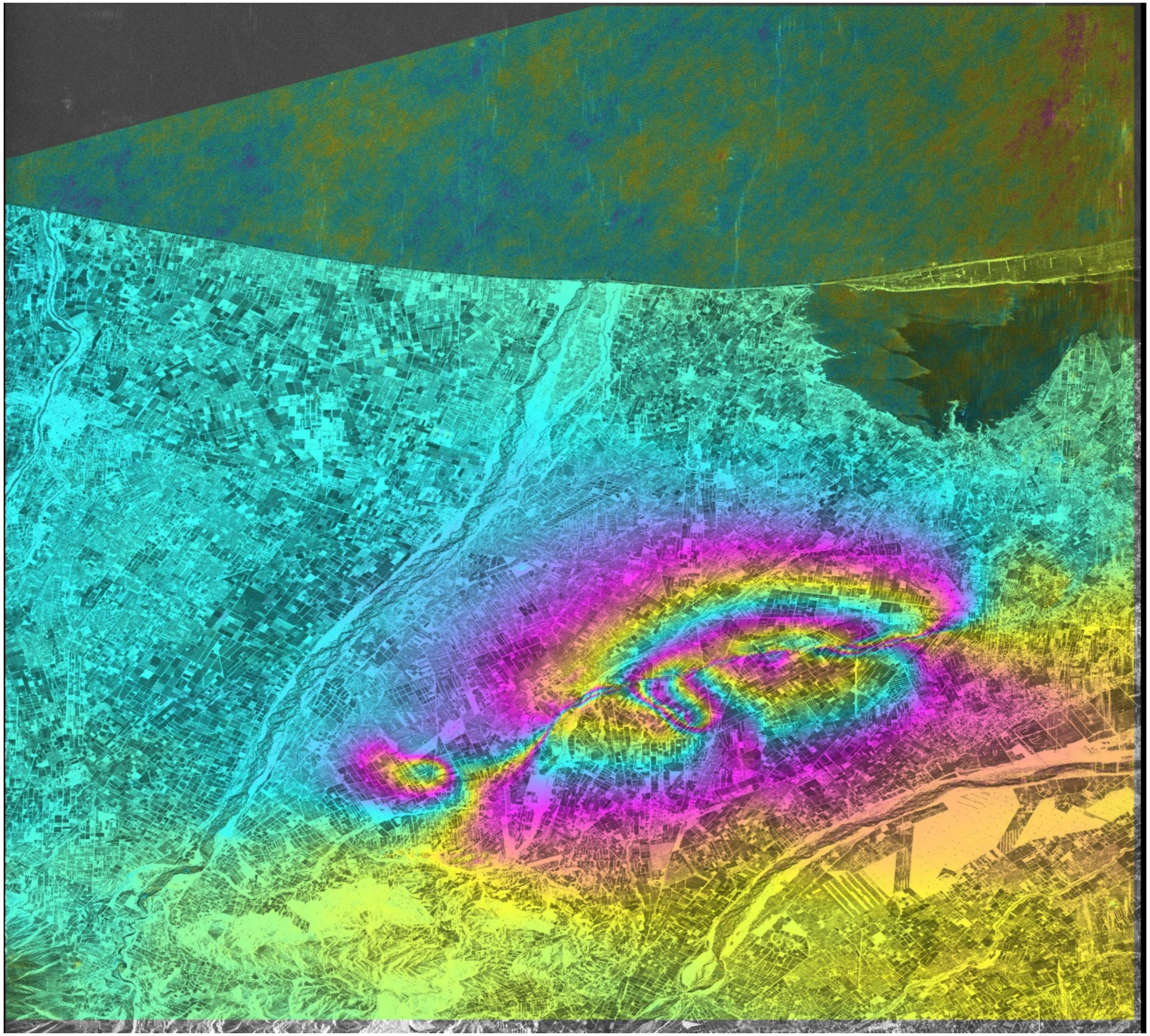
The corresponding color bar can be generated with the program ras8_color_scale of the DISP module

ras8_color_scale colors.ras 1 0. 300. .5 .75

• The color bar is saved to the file colors.ras

• For interferometric products, a double hexagon color model is generally used (parameter value 1).

Displacement map



Displacement map in slant range coordinates, phase-colour-coded with SAR intensity image in the background.

Some notes on errors in interferograms

Besides the quantity of interest an elevation map or displacement map can present errors due to

- Atmospheric heterogeneities
- Phase unwrapping errors
- Baseline errors
- SAR Processing errors (e.g. wrong Doppler ambiguity)
- DEM errors (for differential interferometry)

Separation of errors from true measurement of elevation/displacement requires iteration of several steps that include

- baseline improvement
- estimation and reduction of quadratic phase terms
- band-pass filtering (for estimating the atmospheric component)
- eventually masking etc.

Some notes on errors in interferograms

There is not a rule that can be considered general for all interferograms.

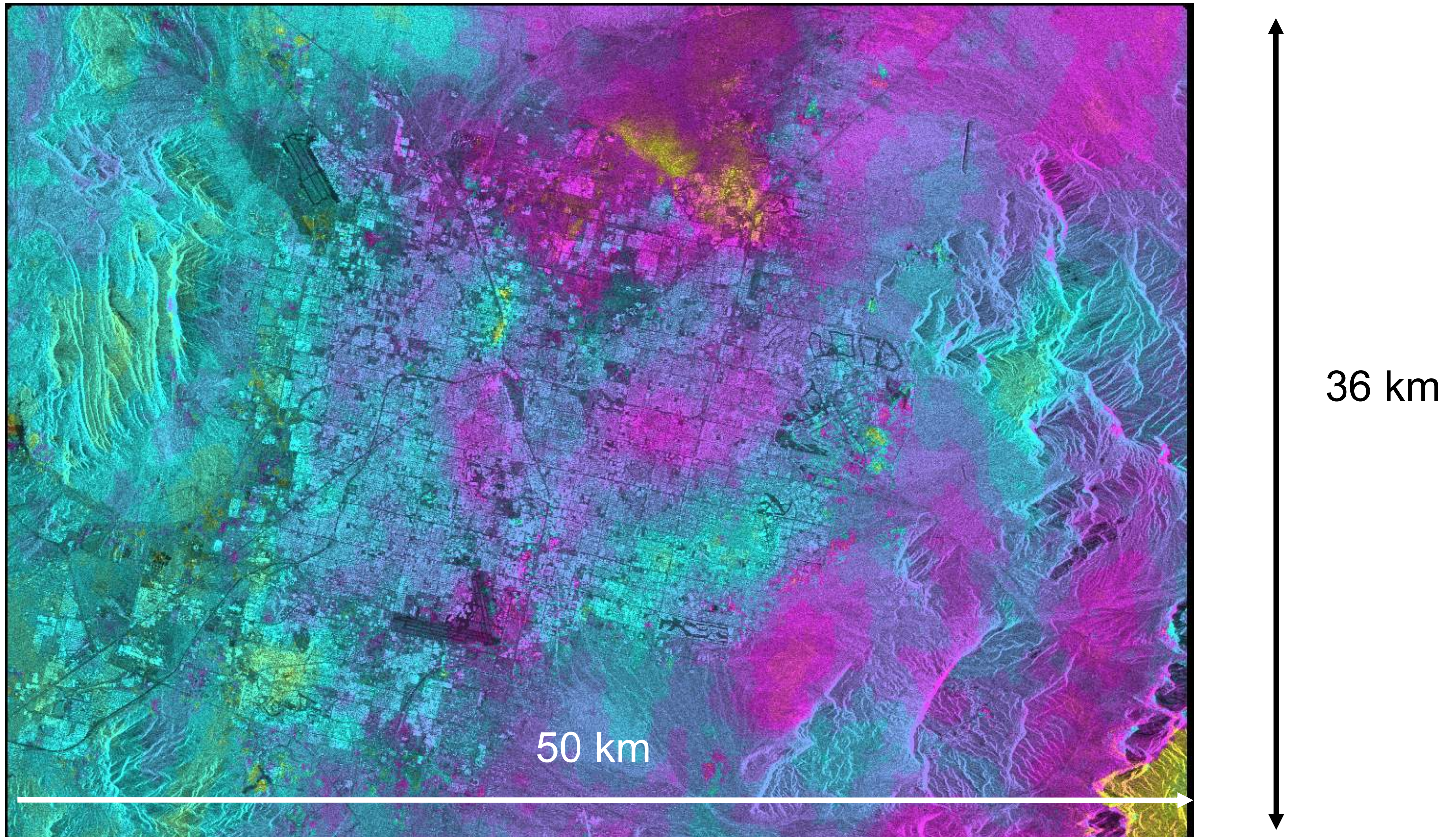
The solution to be applied is related to the specific case

Understanding error sources helps identifying them in an interferogram

and allows finding the best solution to reduce them

Let’s look at two possible examples

Some notes on errors in interferograms: example 1



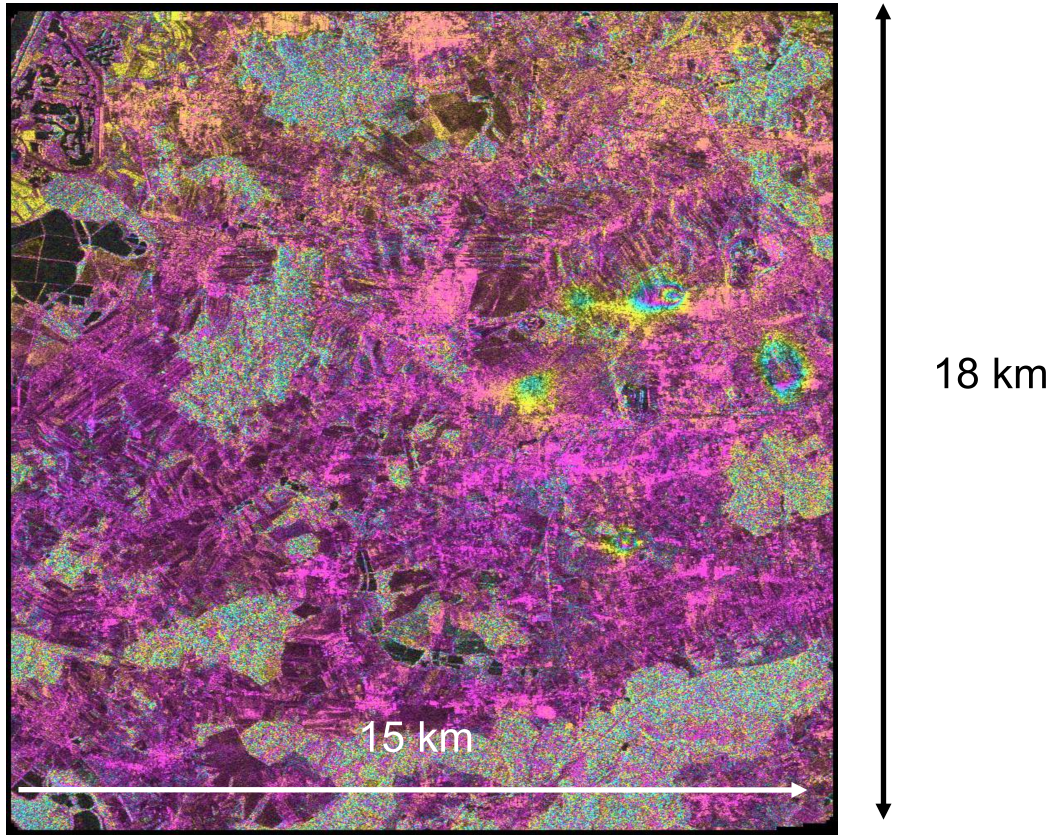
Unwrapped ERS differential interferogram over Las Vegas (Delta_t = 2 years). Known phenomena: local deformation due to subsidence in the city

Small blobs with homogeneous phase → not everything is displacement! →

very likely atmospheric errors are dominant →

Possible solution(s): low-pass filter or stacking

Some notes on errors in interferograms: example 2



Unwrapped ALOS PALSAR differential interferogram over mining area in Poland. Subsidence is expected locally where extraction takes place

Slight phase variation along the vertical direction → this is not displacement! →

Large-scale atmospheric effect and/or baseline error →

Possible solution(s): baseline improvement or estimation of quadratic phase trend

Overview

- Interferometric processing
- Introduction to the Interferometric SAR Processor (ISP) software
- Basic elements of ISP programs
- Use of ISP for processing
- Summary