# Geocoding

# Geocoding with the DIFF&GEO module

*Gamma Remote Sensing AG*

*Worbstrasse 225*

*3073 Gümligen*

*Switzerland*

*santoro@gamma-rs.ch*

*http://www.gamma-rs.ch*

*(version: 2017-11-17)*

## Scope of presentation

- Introduce to geocoding of SAR data. Geocoding consists of a transformation between the geometry of the radar image and the geometry on ground or vice versa. Here, we describe the procedure implemented in the DIFF&GEO Module of the GAMMA software and show how geocoding can be done with the ALOS PALSAR Fine Beam dataset. The image acquired on 13 August 2010 will be processed interactively. Some of the results will be used for interferometric processing.

# Overview

- Meaning of geocoding, DEM
- Image geocoding procedure
- Introduction to the DIFF&GEO module for geocoding
- Geocoding with the DIFF&GEO module
- Summary

# Importance of terrain geocoding

- Terrain geocoding is an essential element for remote sensing analysis in a heterogeneous sensor environment.

- Required for combining data acquired with different sensor modes and imaging geometries.

- Required to combine data acquired using different imaging systems (optical/radar)

- Necessary to be able to perform differential SAR interferometry

# Geocoding: what is it?

Geocoding means in a broad sense the transformation of coordinates from a SAR reference system to a map, i.e. cartographic, reference system.

Geocoding of SAR data takes the SAR image and resamples the it into a map projection (UTM, Oblique-Mercator, Polar Stereographic…) with an associated datum.

- In this sense geocoding is necessary to combine information retrieved by the imaging system (e.g. the SAR image and products derived from it) with information in map coordinates (e.g. a digital elevation model, a land-use inventory, geocoded information from optical remote sensing, etc.).

Geocoding of an image in map coordinates resamples the image to the slant-range / azimuth geometry of a SAR image.

- Typically in this way a DEM in cartographic coordinates can be expressed in the radar geometry and used for differential interferometry.

# DEM sources

Geocoding typically is aided by a Digital Elevation Model (DEM). DEMs can be available from many sources (satellite data, ground survey, topographic maps etc.)

Two (nearly) global datasets freely available that can be used for geocoding are:

GTOPO30 DEM

- 30 arcsec (~ 1 km) resolution
- global
- available at http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html

SRTM-3 DEM

- 3 arcsec (~ 90 m) resolution
- for latitudes between 60 deg N and 60 deg S
- Version 2 available at http://www2.jpl.nasa.gov/srtm/
- Version 4 (void filled) available from CGIAR at http://srtm.csi.cgiar.org/

A merged global DEM from SRTM-3 and GTOPO30 is also available

High resolution SRTM-1 DEM (~ 30 m) is also available but locally

All datasets are in equiangular (latitude/longitude) projection

# Overview

- Meaning of geocoding, DEM
- Image geocoding procedure
- Introduction to the DIFF&GEO module for geocoding
- Geocoding with the DIFF&GEO module
- Summary

# Geocoding: procedure

- The three main steps required in geocoding are

  (i) the initial determination of the geometric transformation,

  (ii) the refinement of the geometric transformation, and

  (iii) the resampling of the image(s) from one coordinate system to the other

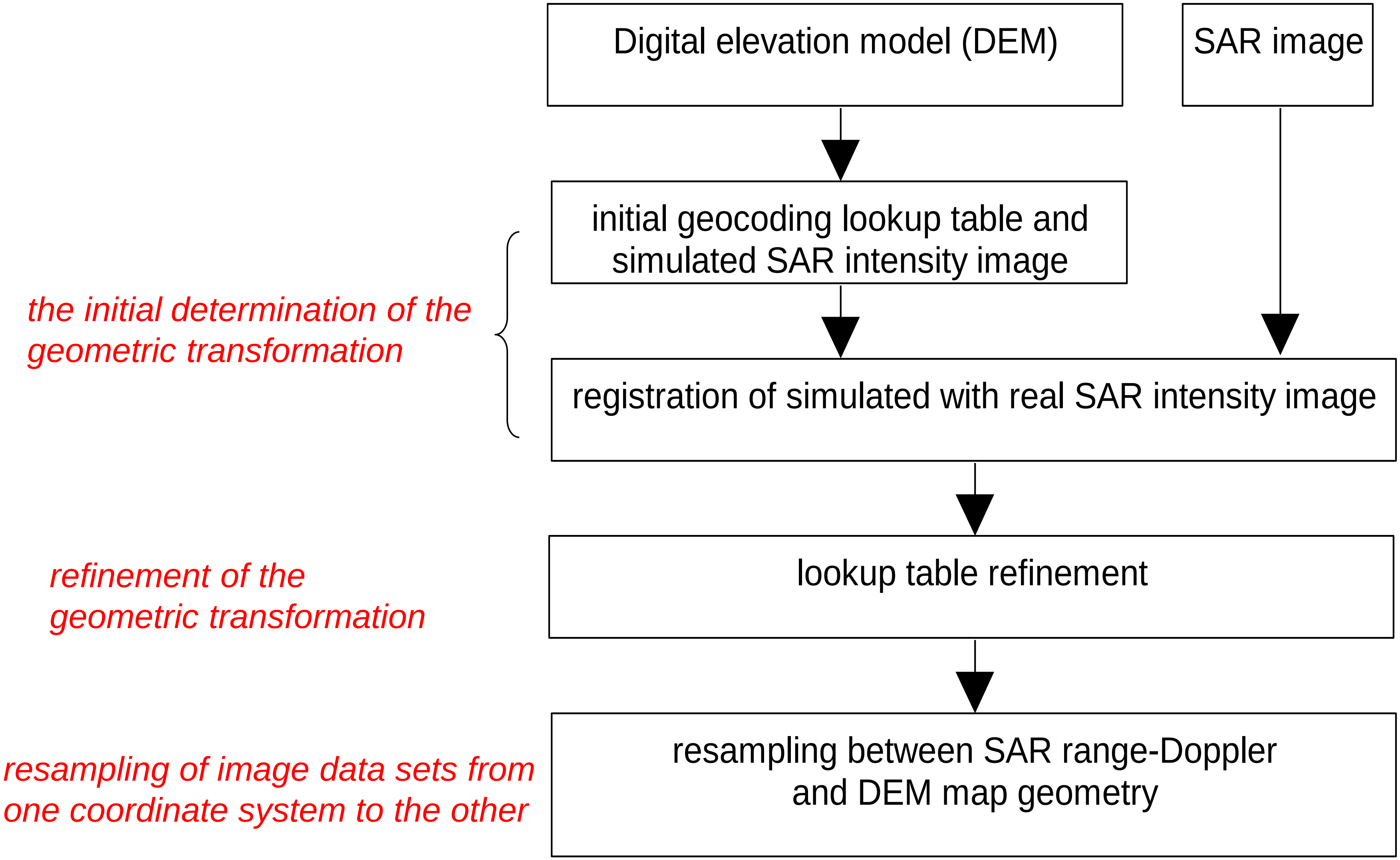
- The geocoding system must be able to correct residual errors due to:

  - Orbit state vectors and interpolation error

  - Processing system errors (focusing/geolocation due to velocity errors)

- Operate with a minimum of operator intervention. Typical approach uses ground control points. A more advanced approach uses a lookup table. This is implemented in the geocoding algorithm of the GAMMA software

# Geocoding – data flow of GAMMA software



# Geocoding lookup table

- The lookup table contains for each image point defined in one coordinate system the corresponding coordinates in the other coordinate system.

- The initial geocoding lookup table is calculated based on the heights in map coordinates and parameters describing the SAR imaging geometry.

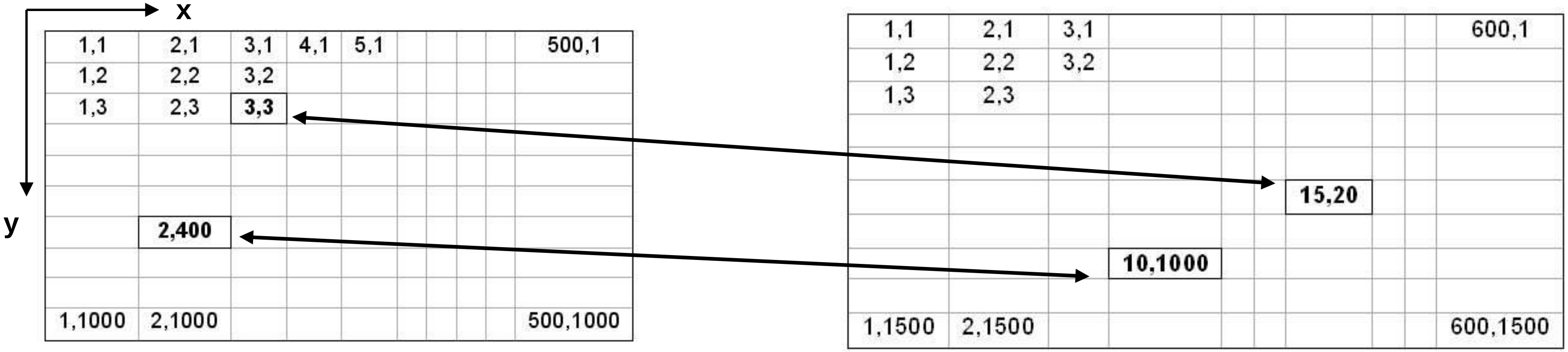
- The lookup table contains complex numbers and has the dimension of data files in map coordinates. The real and the imaginary part represent row and column of pixel at that location in the SAR image.

- The lookup table can be applied to convert from radar to map geometry (so called “backward transformation”) or from map to radar geometry (so called “forward transformation”)

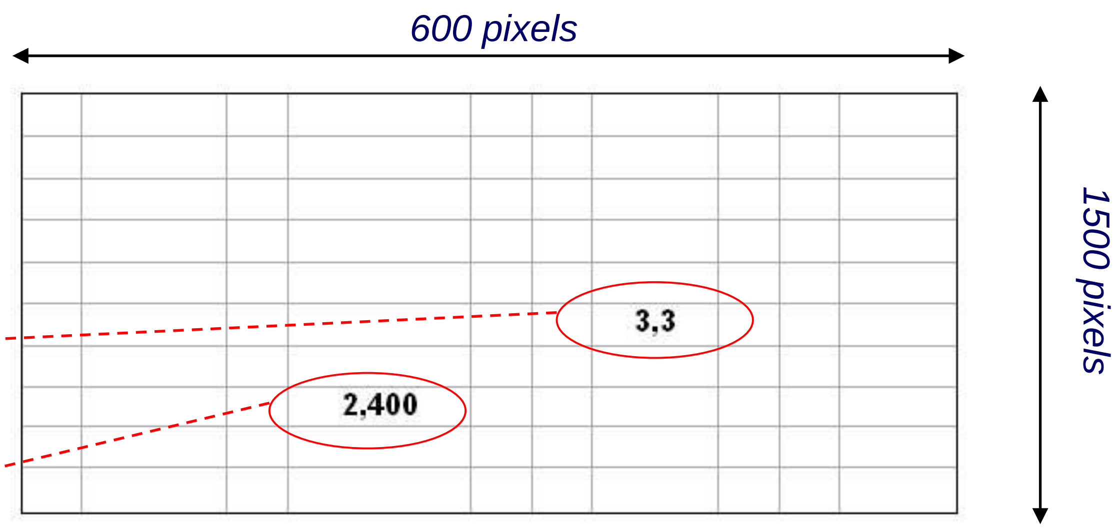
# Example of lookup table

*Pixel positions in SAR image,*
*500 pixels wide and 1000 pixels long*

*Pixel positions in DEM image,*
*600 pixels wide and 1500 pixels long*




E.g. the position (x,y)= (2,400) in the SAR image corresponds to the position (x,y)=(10,1000) in the DEM




**Look-up table** for the transformation between SAR and map geometry

*Value at row 20 and column 15*

*Value at row 1000 and column 10*

# Issue with lookup table

- The initial lookup table is not completely accurate because of limitations in the knowledge of the SAR imaging geometry (the orbit positions are only known with a certain accuracy). As a result geocoding with the initial lookup table leads to geocoding errors of the order of a few pixels, depending mainly on the accuracy of the orbit data.

- For this reason refinement of the lookup table is necessary

# Lookup table refinement

- In the refinement step offsets (=shifts) are determined between corresponding locations in the SAR image and a dataset that can be considered as reference for the output geometry.

- One way to determine such offsets is visual interpretation of the geocoded SAR image and a topographic map, i.e. use of ground control points

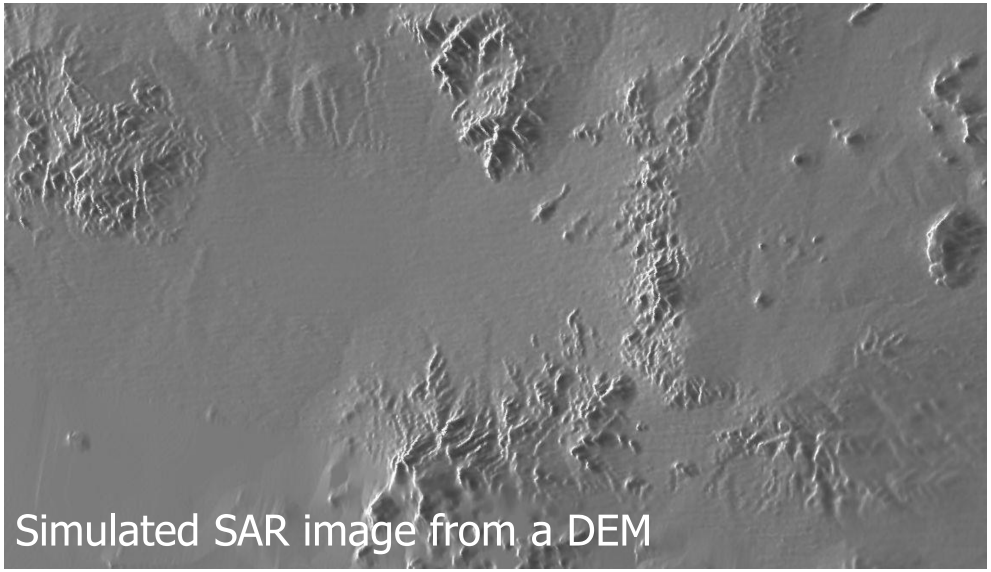
- A more automatic method consists in the comparison between the SAR image and a simulated SAR image based on the DEM. The registration offsets between the simulated SAR image and the true SAR image are determined automatically using correlation analysis. This refinement step requires some terrain features that function as ground control.

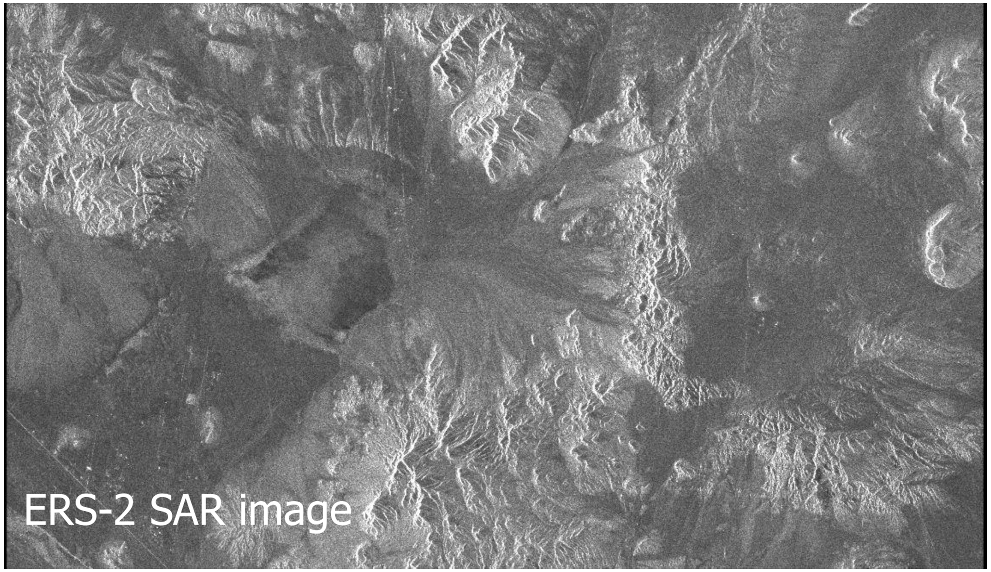
- Either approach results in registration offsets which are then modeled as a bi-linear function of range and azimuth.

- This function is then applied to the initial geocoding lookup table to result in the refined geocoding lookup table.

# Example of simulated and true SAR images

The advantage of using as reference a simulated SAR image from a DEM is that it has similar features compared to the SAR image of the same area


Simulated SAR image from a DEM


ERS-2 SAR image

# Resampling of an image

- The refined geocoding lookup table is used to geocode data from one coordinate system into the other coordinate system in one step.

- In the case of a SAR image in the radar geometry, this means that the image is resampled to map geometry (backward transformation)

- In the case of a DEM in map coordinates, this means that the DEM is resampled to the SAR geometry to a height map (forward transformation)

# Note on resampling

For the resampling step it is recommended that the spatial resolution of the image remains about the same. A practical reason for this is ensure that the sampling is adequate for the interpolator. But there are also other reasons.

***Problem***: If we have a SAR image at higher resolution and want to geocode it to a **lower resolution** , we would have the problem that the noise is kept unnecessarily high.
***Solution***: first apply multi-looking (or filtering) to reduce the spatial resolution to a similar value as required for the geocoded image and then geocode.

***Problem***: If we have a SAR image at **lower resolution** and want to geocode it to a higher resolution, the geocoding efficiency would be much reduced without much information gain.
***Solution***: geocode at a lower resolution and then oversample the geocoded image to the required higher resolution.

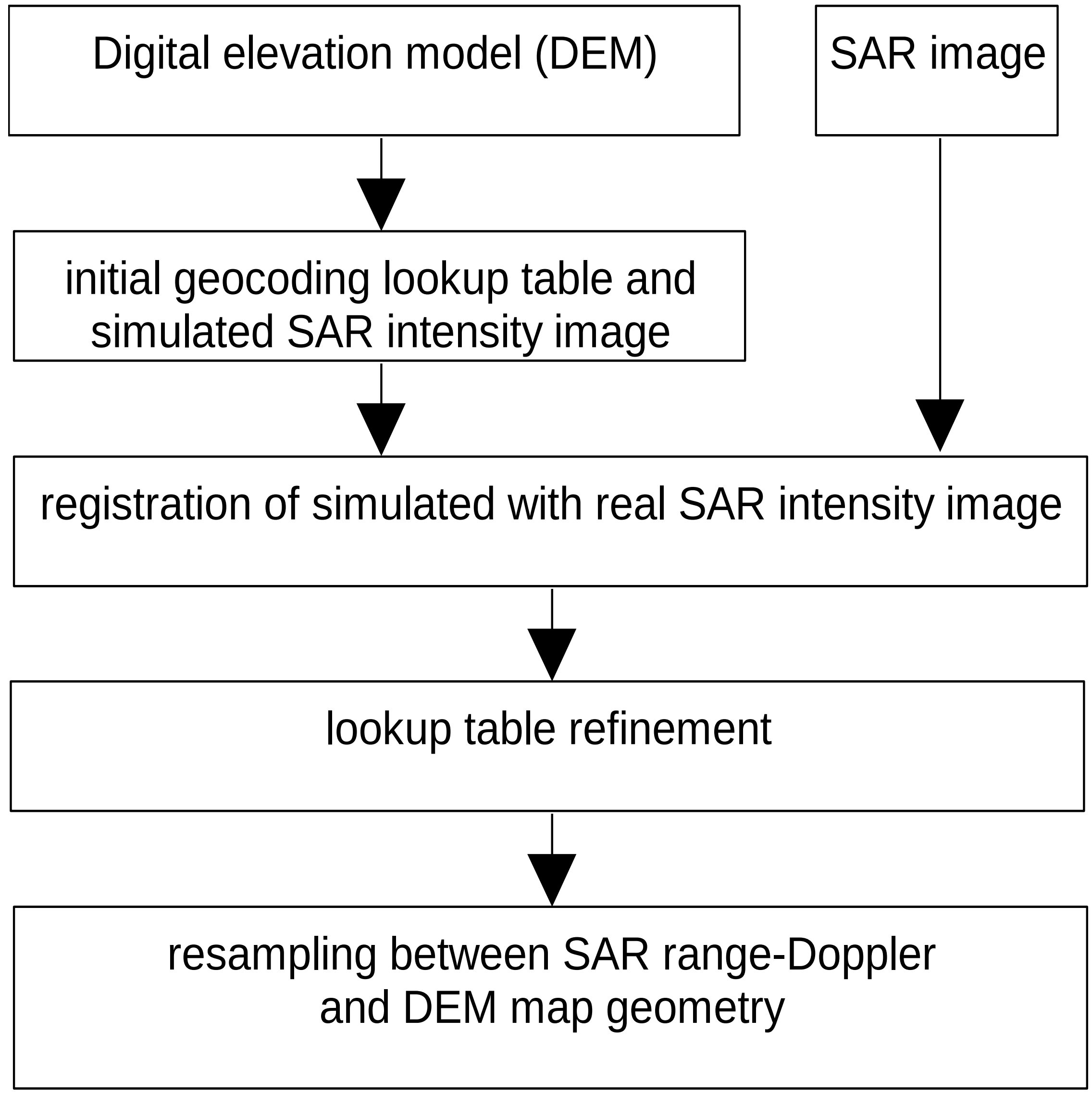
**Q**: If I have a SAR image with a 20x20 m pixel size and the DEM for geocoding is 50x50 m what should I do?

**A**: Multi-look the SAR image for example with factors 2 both in range and in azimuth to obtain pixels of 40 x 40 m and then geocode this image.

# Overview

- Meaning of geocoding, DEM
- Image geocoding procedure
- Introduction to the DIFF&GEO module for geocoding
- Geocoding with the DIFF&GEO module
- Summary

# Geocoding with DIFF&GEO



# What can DIFF&GEO do for geocoding?

The module encompasses a full range of algorithms required for

- direct geocoding of SAR images
- inverse transformation of maps to SAR geometry

With DIFF&GEO the following steps for geocoding a SAR image are performed

- pre-processing, generation of parameter files
- pre-processing, preparation of DEM and related parameter file
- generation of an initial lookup table
- refinement of lookup table (manual or automatic via simulated SAR image)
- resampling of SAR image to map coordinates

Processing related parameters and data characteristics are saved as text files

# Overview

- Meaning of geocoding, DEM
- Image geocoding procedure
- Introduction to the DIFF&GEO module for geocoding
- Geocoding with the DIFF&GEO module
- Summary

# Dataset

SLC image (generated with the MSP module) acquired by ALOS PALSAR over Christchurch, New Zealand, on 13 August 2010

ALOS PALSAR input image: 20100813.slc

For image geoding, the following data is furthermore needed:

- the SLC parameter file containing all important information concerning the image (size, acquisition time, geometry) and the satellite
- a DEM of the area covered by the SAR image

**In this exercise, we will**

1) Prepare the images for geocoding (SAR intensity image, DEM)

2) Compute the initial version of the lookup table

3) Refine the lookup table

4) Resample the intensity image from SAR to map coordinates

5) Resample the DEM from map to SAR coordinates (for interferometry on day 3)

6) Generate a kml file of geocoded SAR image to be displayed in Google Earth

# Input dataset

The dataset consists of :

- SLC image: 20100813.slc
- SLC parameter file: 20100813.slc.par
- DEM image: NZ.EQA.dem
- Parameter file of DEM image: NZ.EQA.dem_par

Before geocoding, we need to learn how to

- obtain an MLI (intensity) image from the SLC image.
- understand the meaning of the parameter file for the DEM
- display the DEM

NOTE: from now onwards

- input file names are displayed in green

- output file names are displayed in red

- File names that are used in input and are modified by the program are displayed in brown

# Generation of image and parameter file from original data

| Program: | Sensor | Image Type and Facility |
|---|---|---|
| par_ACS_ERS | ERS | SLC data from the ACS processor used by Indian PAF |
| par_ASAR | ENVISAT ASAR | SLC (Alternating Polarization, Image Mode, Wide Swath modes) and PRI data (also Global Monitoring mode) |
| par_ASF_91 | various | SLC data from Alaskan SAR Facility, Fairbanks (1991-1996) |
| par_ASF_96 | various | SLC data from Alaskan SAR Facility, Fairbanks (after 1996) |
| par_ASF_PRI | various | Ground range detected images produced by the Alaskan SAR Facility, Fairbanks after 1996, available for ERS-1, ERS-2, Radarsat-1, and JERS-1 |
| par_ASF_RSAT_SS | RADARSAT-1 | Ground range detected SCANSAR images produced by the Alaskan SAR Facility |
| par_ATLSCI_ERS | ERS | SLC data produced using the Atlantis APP processor (CCRS) |
| par_CS_SLC | COSMO-SkyMed | COSMO-Skymed SLC data in HDF5 format produced by ASI |
| par_CS_SLC_TIF | COSMO-SkyMed | COSMO-Skymed SLC data in TIF format produced by ASI |
| par_EORC_JERS_SLC | JERS | SLC data produced by JAXA EORC |
| par_EORC_PALSAR | ALOS PALSAR | Level 1.1 SLC data produced by JAXA EORC |
| par_ERSDAC_PALSAR | JERS | Level 1.1 SLC data produced by ERSDAC |
| par_ESA_ERS | ERS | SLC data from the German, ASI, UK or ESRIN PAF, either VMP or PGS processed data |
| par_IECAS_SLC | IECAS SAR | Generate ISP image parameter file and SLC image in fcomplex format |
| par_KC_PALSAR_slr | ALOS PALSAR | Detected SAR intensity data strips in slant range geometry produced by JAXA EORC for the Kyoto and Carbon Initiative |
| par_KS_SLC | KOMPSAT-5 | Kompsat-5 SCS (Level 1A) data as provided by KARI |
| par_MSP | various | SLC data processed with the Gamma MSP module |
| par_PRI | ERS | PRI ground range data produced by ESA PAFs |
| par_PRI_ESRIN_JERS | JERS-1 | PRI ground range data produced by ESA ESRIN |
| par_PulSAR | ERS | SLC data processed using the PulSAR SAR processor from Phoenix Systems |
| par_RISAT_GRD | RISAT-1 | Read RISAT-1 Ground-Range data from a CEOS data set and perform radiometric calibration |
| par_RISAT_SLC | RISAT-1 | Read RISAT-1 CEOS format SLC data and perform radiometric calibration |
| par_RSAT_SCW | RADARSAT-1 | Wide-swath SCANSAR images (8-bit/value ), |
| par_RSAT_SGF | RADARSAT-1 | Radarsat stripmap path images (SGF) are analogous to ESA ERS PRI images |
| par_RSAT_SLC | RADARSAT-1 | Radarsat strip map SLC data processed by Radarsat International (RSI), Atlantis APP SAR processor, or the ASF BPP processor |
| par_RSAT2_SG | RADARSAT-2 | Radarsat ground range SGF/SGX data processed by Radarsat International (RSI) |
| par_RSAT2_SLC | RADARSAT-2 | Radarsat strip map SLC data processed by Radarsat International (RSI) |
| par_RSI_ERS | ERS | SLC data processed by Radarsat International (RSI) |
| par_SIRC | SIR-C | SLC data from JPL or USGS EOC |
| par_TX_GRD | TerraSAR-X | MGD, i.e. PRI, ground range data processed by DLR |
| par_TX_SLC | TerraSAR-X | SSC, i.e. SLC, data processed by DLR |
| par_UAVSAR_SLC | UAVSAR | MLC,SLC data products processed by JPL |

**EXAMPLES:**

```
par_ESA_ERS LEA_01.001 19950922.par DAT_01.001 19950922.slc
```

# How to view a parameter file

To view any parameter file created by the programs of the GAMMA software

- Open the file with a text editor or

- Type at the command line

more *name_of_parameter_file*

For example we can look at the content of the parameter file of the SLC with:

more 20100813.slc.par

(NOTE: “more” displays the part of the file that fits the terminal window. If the text is longer, press the space bar to scroll down)

# The ISP Image Parameter file (1/2)

```
Gamma Interferometric SAR Processor (ISP) - Image Parameter File
title:     SAR
sensor:    PALSAR
date:      2010    8   13
start_time:            43086.962527   s
center_time:           43093.406139   s
end_time:              43099.849752   s
azimuth_line_time:     4.7500000e-04  s
line_header_size:                 0
range_samples:                 5016
azimuth_lines:                27132
range_looks:                      1
azimuth_looks:                    1
image_format:              SCOMPLEX
image_geometry:            SLANT_RANGE
range_scale_factor:     1.0000000e+00
azimuth_scale_factor:   1.0000000e+00
center_latitude:        -43.7279320   degrees
center_longitude:       172.1215280   degrees
heading:                -18.1526580   degrees
range_pixel_spacing:        9.368514   m
azimuth_pixel_spacing:      3.251988   m
near_range_slc:          864366.9634   m
center_range_slc:        887858.5122   m
far_range_slc:           911350.0611   m
first_slant_range_polynomial:       0.00000      0.00000  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  s m 1 m^-1 m^-2 m^-3
center_slant_range_polynomial:      0.00000      0.00000  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  s m 1 m^-1 m^-2 m^-3
last_slant_range_polynomial:        0.00000      0.00000  0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  s m 1 m^-1 m^-2 m^-3
incidence_angle:            39.0022   degrees
azimuth_deskew:          ON
azimuth_angle:              90.0000   degrees
radar_frequency:       1.2700000e+09   Hz
adc_sampling_rate:     1.6000000e+07   Hz
chirp_bandwidth:       1.4000000e+07   Hz
prf:                   2105.263158   Hz
azimuth_proc_bandwidth:   1684.21053   Hz
```

# The ISP Image parameter file (2/2)

```
doppler_polynomial:        42.02703  2.70939e-05  0.00000e+00  0.00000e+00  Hz     Hz/m     Hz/m^2     Hz/m^3
doppler_poly_dot:       0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  Hz/s   Hz/s/m   Hz/s/m^2   Hz/s/m^3
doppler_poly_ddot:      0.00000e+00  0.00000e+00  0.00000e+00  0.00000e+00  Hz/s^2 Hz/s^2/m Hz/s^2/m^2 Hz/s^2/m^3
receiver_gain:              24.0000   dB
calibration_gain:            6.0000   dB
sar_to_earth_center:            7080022.4360   m
earth_radius_below_sensor:      6367964.0686   m
earth_semi_major_axis:          6378137.0000   m
earth_semi_minor_axis:          6356752.3141   m
number_of_state_vectors:                  11
time_of_first_state_vector:      42780.000000   s
state_vector_interval:              60.000000   s
state_vector_position_1:  -3181184.1716    240183.8929   -6321280.8687   m   m   m
state_vector_velocity_1:    -6136.17890      3063.14070      3205.81960   m/s m/s m/s
state_vector_position_2:  -3541930.1026    424957.3679   -6116304.8855   m   m   m
state_vector_velocity_2:    -5884.61090      3093.54970      3624.44020   m/s m/s m/s
state_vector_position_3:  -3886859.9263    611115.5492   -5886637.4811   m   m   m
state_vector_velocity_3:    -5609.16850      3109.24550      4028.59060   m/s m/s m/s
state_vector_position_4:  -4214576.8983    797766.9570   -5633196.0666   m   m   m
state_vector_velocity_4:    -5311.05810      3109.94990      4416.64020   m/s m/s m/s
state_vector_position_5:  -4523759.4819    984005.4909   -5356994.1697   m   m   m
state_vector_velocity_5:    -4991.57980      3095.45430      4787.01650   m/s m/s m/s
state_vector_position_6:  -4813166.8679   1168914.7071   -5059137.7471   m   m   m
state_vector_velocity_6:    -4652.12370      3065.62290      5138.21240   m/s m/s m/s
state_vector_position_7:  -5081644.2041   1351572.2435   -4740821.0261   m   m   m
state_vector_velocity_7:    -4294.16380      3020.39440      5468.79380   m/s m/s m/s
state_vector_position_8:  -5328127.4262   1531054.3226   -4403321.9294   m   m   m
state_vector_velocity_8:    -3919.25120      2959.78300      5777.40600   m/s m/s m/s
state_vector_position_9:  -5551647.7018   1706440.2941   -4047997.1654   m   m   m
state_vector_velocity_9:    -3529.00830      2883.87890      6062.77830   m/s m/s m/s
state_vector_position_10:  -5751335.5133   1876817.2260   -3676276.9880   m   m   m
state_vector_velocity_10:    -3125.12200      2792.84920      6323.73090   m/s m/s m/s
state_vector_position_11:  -5926424.3467   2041284.5218   -3289659.6052   m   m   m
state_vector_velocity_11:    -2709.33760      2686.93810      6559.17950   m/s m/s m/s
```

# Generation of SAR intensity image

The intensity (MLI) image is obtained from the SLC image with the program of the ISP module multi_look

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

The program generates an MLI image and a corresponding parameter file. The format of this parameter file is identical to the SLC parameter file given as input.

As multi-look factors we used 2 (range) and 12 (azimuth).

| | | | |
|---|---|---|---|
| *SLC range sample spacing (m):* | *9.36851* | *MLI range sample spacing (m):* | *18.73703* |
| *SLC azimuth sample spacing (m):* | *3.25199* | *MLI azimuth sample spacing (m):* | *39.02386* |
| *SLC image range samples:* | *5016* | *MLI image range samples:* | *2508* |
| *SLC image azimuth lines:* | *27132* | *MLI image azimuth lines:* | *2261* |

This corresponds to roughly 30 m on ground

**Ground range spacing = range spacing / sine (incidence angle)**

*Here incidence angle is 39 degrees*

# Display of SAR intensity image

The MLI image can be displayed with the program dispwr of the DISP module

dispwr 20100813.mli 2508

*Multi-look image 20100813.mli*

# Explanation of DEM parameter file

- A DEM parameter file is a text file that describes an image in a map coordinate system. It contains information about size of the image, geographic location, pixel size, map projection and datum.
- A DEM parameter file is not strictly necessary for a DEM but also applies to other images.
- A DEM parameter file is generated with the program create_dem_par
- The program is interactive and requires the user to input several information.
- For this exercise, a DEM parameter file for a Digital Elevation Model has been provided (file NZ.EQA.dem_par).
- We can take a look at it by typing

  more NZ.EQA.dem_par

# DEM parameter file

```
Gamma DIFF&GEO DEM/MAP parameter file
title: DEM_P337
DEM_projection:     EQA
data_format:        REAL*4
DEM_hgt_offset:          0.00000
DEM_scale:               1.00000
width:              3640
nlines:             4480
corner_lat:   -42.7556000  decimal degrees
corner_lon:    171.2580000  decimal degrees
post_lat:  -3.0000000e-04  decimal degrees
post_lon:    4.0000000e-04  decimal degrees

ellipsoid_name: WGS 84
ellipsoid_ra:        6378137.000   m
ellipsoid_reciprocal_flattening:  298.2572236

datum_name: WGS 1984
datum_shift_dx:            0.000   m
datum_shift_dy:            0.000   m
datum_shift_dz:            0.000   m
datum_scale_m:         0.00000e+00
datum_rotation_alpha:  0.00000e+00   arc-sec
datum_rotation_beta:   0.00000e+00   arc-sec
datum_rotation_gamma:  0.00000e+00   arc-sec
datum_country_list Global Definition, WGS84, World
```

*Map projection (EQA= lat/long)*

*Size of corresponding image*

*Coordinates of top-left pixel (centre of pixel)*

*Pixel size (posting)*

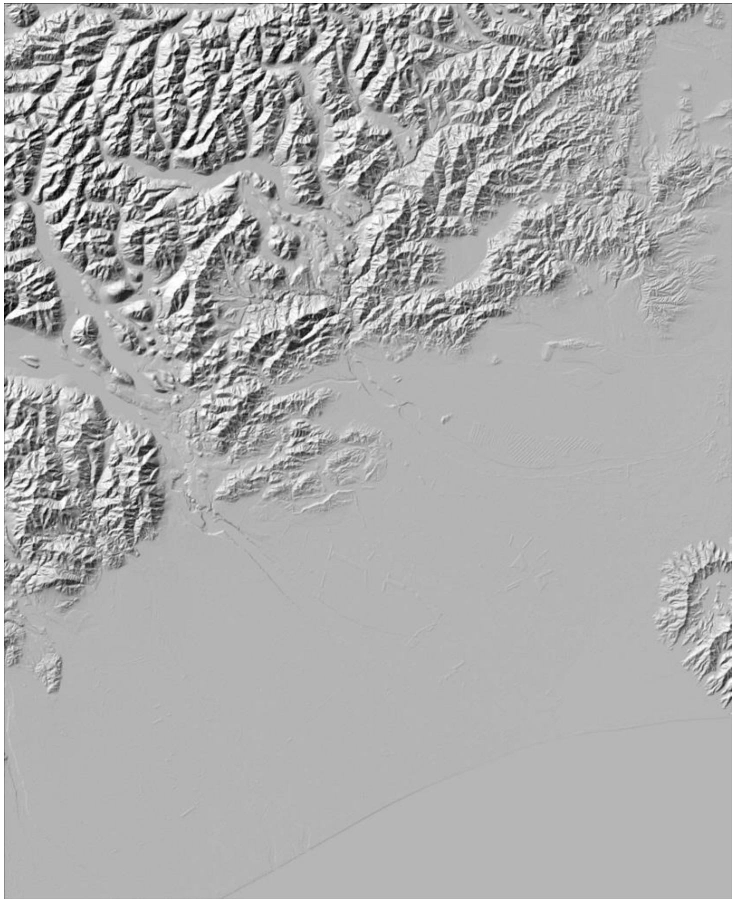
*Approximately 30 m*

*Convention of software: vertical coordinate always negative*

# Display of DEM

The MLI image can be displayed with the program disdem_par of the DISP module. The DEM is displayed in the form of shaded relief. With this program, also the geographic coordinates will be visible.

disdem_par NZ.EQA.dem NZ.EQA.dem_par



# Lookup table generation

To generate the lookup table, we use the program gc_map

gc_map 20100813.mli.par - NZ.EQA.dem_par NZ.EQA.dem Christchurch.EQA.dem_par Christchurch.EQA.dem 20100813.lt 1 1 20100813.EQA.simsar

Input:

- the parameter file of the intensity image, the DEM in and the corresponding parameter file

Output:

- Part of the DEM covering the image and the corresponding parameter file (Christchurch.EQA.dem, Christchurch.EQA.dem_par)
- Initial lookup table (20100813.lt)
- SAR image simulated from the DEM (20100813.EQA.simsar)

NOTE: the input DEM must cover an area that includes the SAR image to be geocoded. The program gc_map will cut out from this the part coincident with the SAR image.

# DEM parameter files

Let's compare the DEM parameter files (original and produced by gc_map)

```
Gamma DIFF&GEO DEM/MAP parameter file
title: DEM_P337
DEM_projection:     EQA
data_format:        REAL*4
DEM_hgt_offset:          0.00000
DEM_scale:               1.00000
width:                 3640
nlines:                4480
corner_lat:     -42.7556000  decimal degrees
corner_lon:     171.2580000  decimal degrees
post_lat:    -3.0000000e-04  decimal degrees
post_lon:     4.0000000e-04  decimal degrees

ellipsoid_name: WGS 84
ellipsoid_ra:        6378137.000   m
ellipsoid_reciprocal_flattening:  298.2572236

datum_name: WGS 1984
datum_shift_dx:              0.000   m
datum_shift_dy:              0.000   m
datum_shift_dz:              0.000   m
datum_scale_m:         0.00000e+00
datum_rotation_alpha:  0.00000e+00   arc-sec
datum_rotation_beta:   0.00000e+00   arc-sec
datum_rotation_gamma:  0.00000e+00   arc-sec
datum_country_list Global Definition, WGS84, World
```

DEM parameter file of original DEM

file: NZ.EQA.dem_par

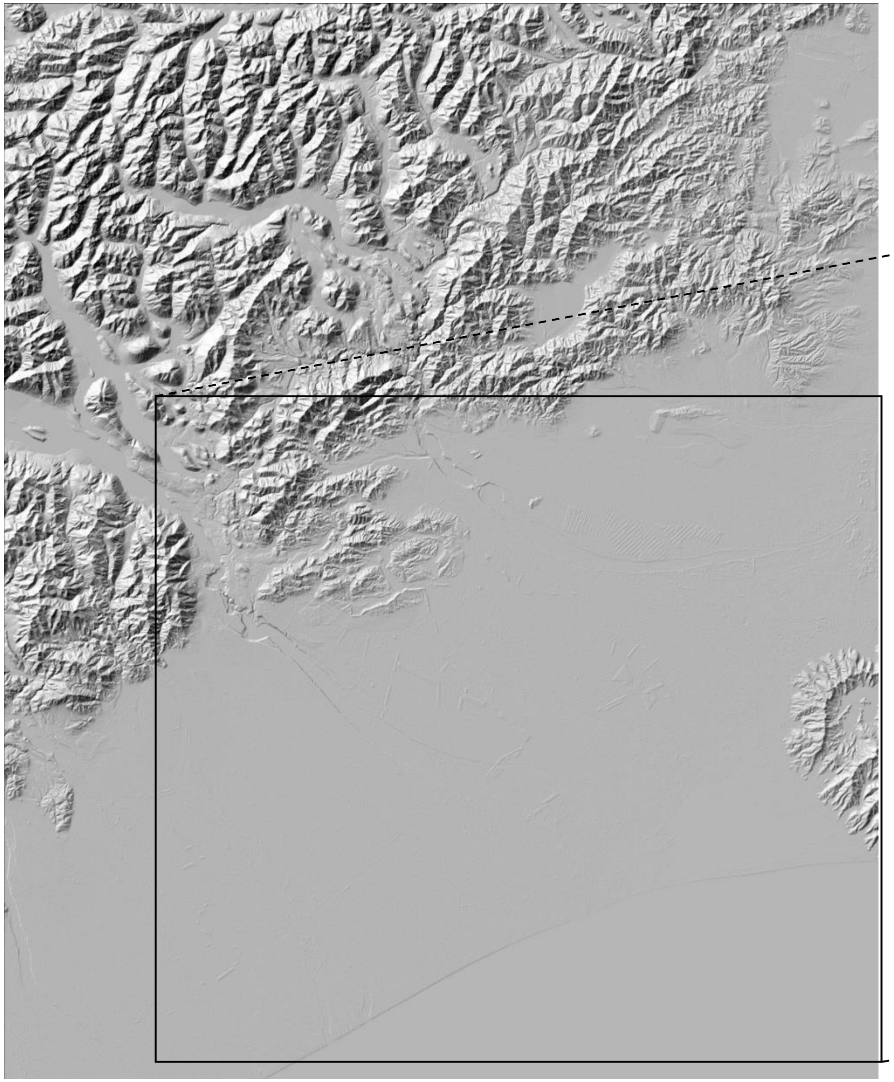
```
Gamma DIFF&GEO DEM/MAP parameter file
title: DEM_P337
DEM_projection:     EQA
data_format:        REAL*4
DEM_hgt_offset:          0.00000
DEM_scale:               1.00000
width:                 2976
nlines:                2852
corner_lat:     -43.2440000  decimal degrees
corner_lon:     171.5236000  decimal degrees
post_lat:    -3.0000000e-04  decimal degrees
post_lon:     4.0000000e-04  decimal degrees

ellipsoid_name: WGS 84
ellipsoid_ra:        6378137.000   m
ellipsoid_reciprocal_flattening:  298.2572236

datum_name: WGS 1984
datum_shift_dx:              0.000   m
datum_shift_dy:              0.000   m
datum_shift_dz:              0.000   m
datum_scale_m:         0.00000e+00
datum_rotation_alpha:  0.00000e+00   arc-sec
datum_rotation_beta:   0.00000e+00   arc-sec
datum_rotation_gamma:  0.00000e+00   arc-sec
datum_country_list Global Definition, WGS84, World
```

DEM parameter file of output DEM

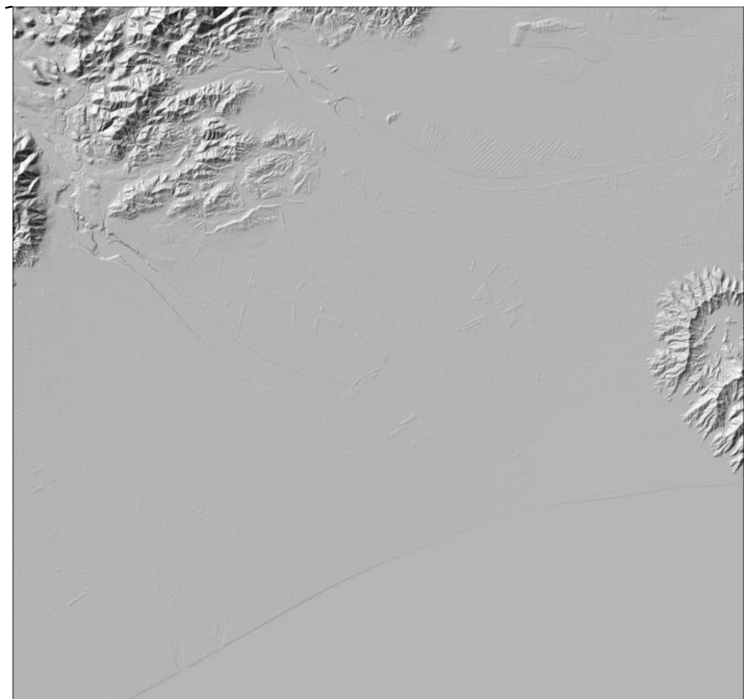
File: Christchurch.EQA.dem_par

# Comparison of DEMs

Original DEM



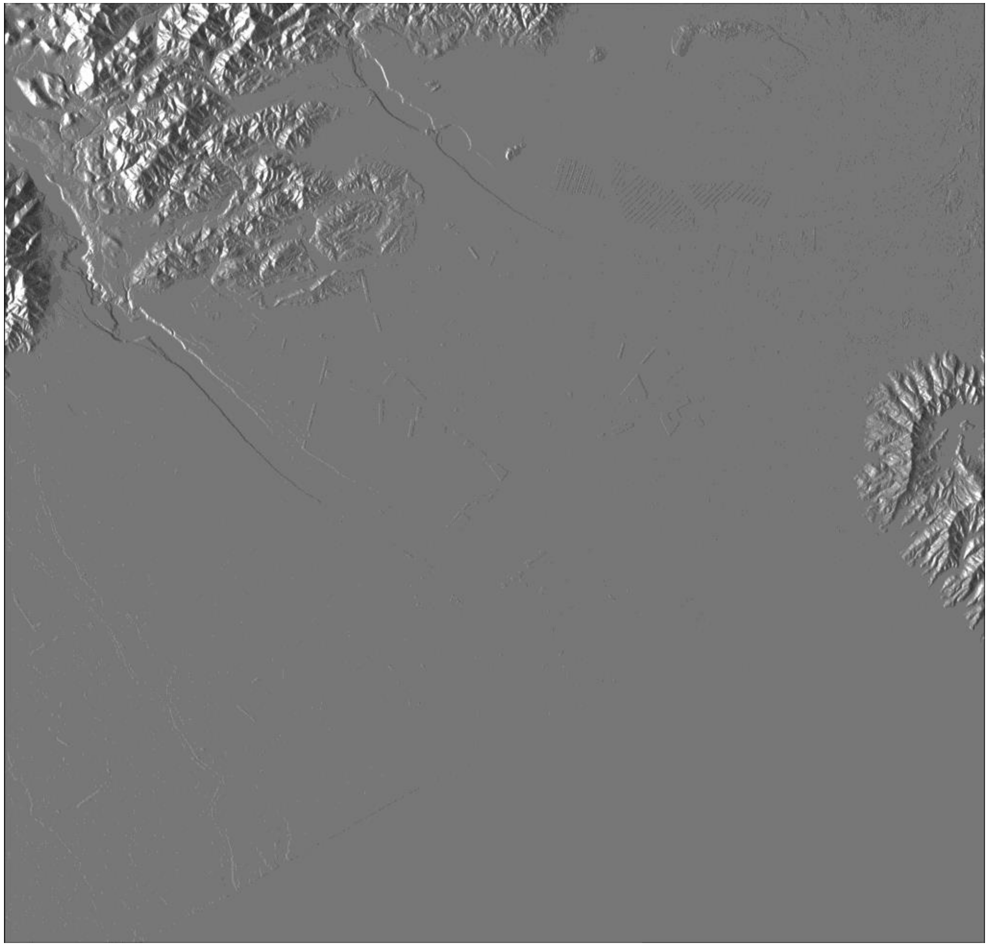
Subset of DEM produced by gc_map, to be used for geocoding



# Simulated SAR image

Let's have a look at the simulated SAR image obtained from the DEM. It can be viewed with the program dispwr of the DISP module.

dispwr 20100813.EQA.simsar 2976



*Simulated SAR intensity image obtained from DEM – map geometry*

# Geocoding lookup table

Let's have a look at the geocoding lookup table. It can be viewed with the program dismph of the DISP module.

dismph 20100813.lt 2976

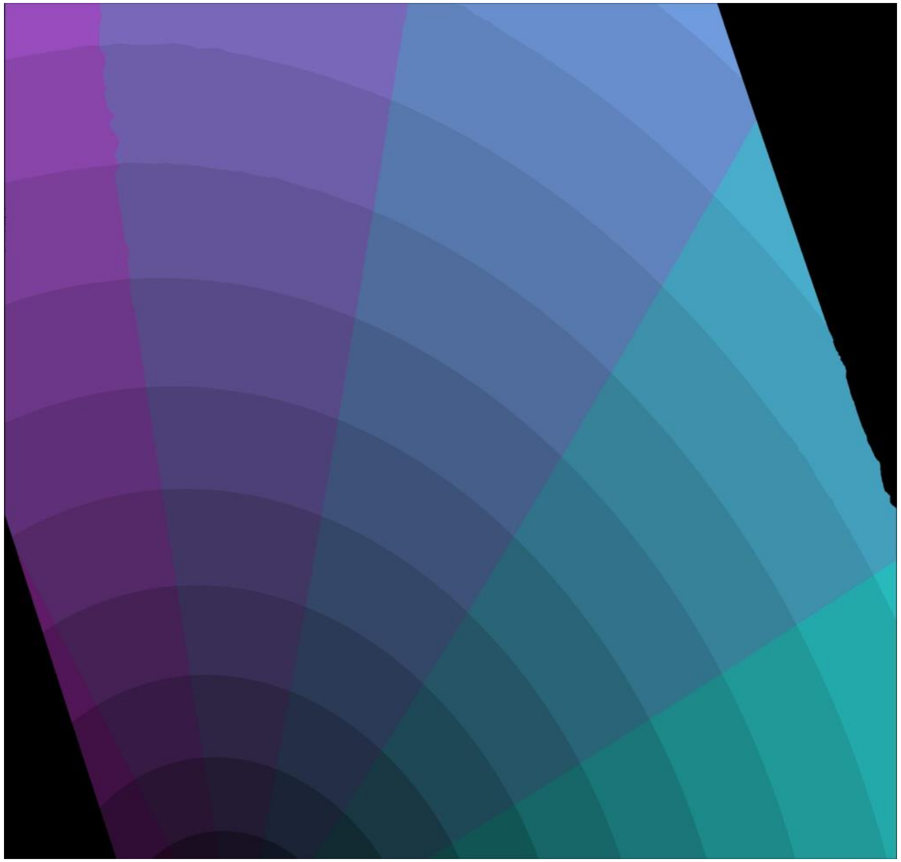


*Image of lookup table. Clicking on a pixel returns the coordinates of the pixel in the radar image (real value = x coordinate, imaginary value = y coordinate)*

# Refinement of lookup table

The lookup table as it is might be affected by location errors.

The steps to refine the lookup table are the following

- Transform the simulated SAR image from map to radar geometry

  (NOTE: at this stage the MLI image and the simulated SAR image are not in the same geometry)

- Compute offsets between simulated SAR image and SAR intensity image
- Determine the polynomial for the registration between the two images
- Update the lookup table using the polynomial

# Refinement of lookup table – transformation of sim sar image

To transform the simulated SAR image from map to SAR geometry we use the program geocode

geocode 20100813.lt 20100813.EQA.simsar 2976 20100813.simsar 2508 2261 0 0

Input

- Initial lookup table
- Simulated SAR image in the map geometry

Output

- Simulated SAR image in the radar geometry

Notice

To process we need to know the number of columns of the simulated SAR in the map geometry (2976) and the dimensions of the image to be obtained in the SAR geometry (2508 x 2261 → the dimensions of the intensity image, stored in the MLI parameter file)

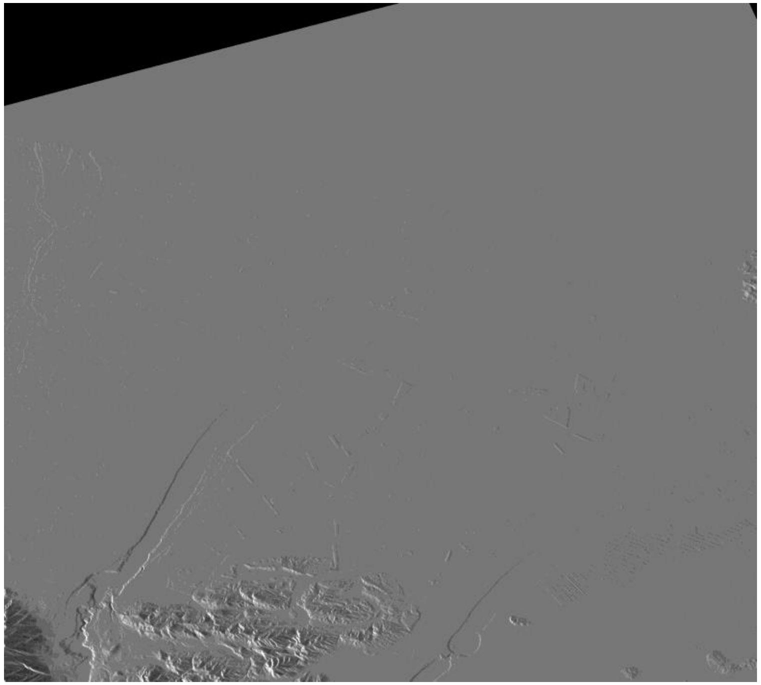
# Comparison of simulated and true SAR images

Let's compare the simulated and the actual SAR images in the radar geometry in order to have a feel for the overlap between the two images. Two images can be displayed simultaneously with the program dis2pwr of the DISP module.

dis2pwr 20100813.simsar 20100813.mli 2508 2508

And we can flip between images

Try to estimate the offset between the images by using topographic features



My guess: Less than a couple of pixels

# Refinement of lookup table - Offset computation procedure

To compute the offsets between the simulated SAR image and the actual SAR we apply the following procedure

1) First of all we need to generate a file (text file) in which the offsets will be stored. To start with we give the offsets the values 0 and 0 in range and azimuth.

2) To find the offsets we take windows all over the images and in each pair of corresponding windows we estimate the offset in range and the offset in azimuth by correlating the intensities.

For the offset computation we apply a two-step procedure

a) First computation of the offsets using few large windows to obtain a first guess of the offsets

b) Refinement of the offsets using more windows of smaller dimension

- At both steps we obtain a "net" of offsets estimates from which the coefficients of the polynomial for the refinement of the look-up table are determined

- Number of windows and their size depend on the specific area

# Generation of offset parameter file (diff_par)

1) Generate a text file in which the offsets will be stored (so called “diff par” file)

For this we use the program create_diff_par that will ask you to input some values

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

To start with we accept all default values.

This file will be updated throughout the refinement step that will be described hereafter

NOTE: A first global estimate of the offsets (constant in range and constant in azimuth) is not done here because not successful due to the mostly flat surface of the area; the program supporting this estimation is called **init_offsetm.**

# Refinement of lookup table - Offset computation (1/2)

We proceed directly with the estimation of local offsets

First computation of offsets using 6 x 6 windows (each 256 x 256 pixels large).

For this we use the program offset_pwrm

offset_pwrm 20100813.simsar 20100813.mli 20100813.diff_par offs snr 256 256 offsets 2 6 6 0.1

The program requires

- The MLI image to be geocoded and the simulated SAR image in radar geometry
- The offset parameter file

The user can select

- The number of windows and size (they are homogeneously distributed over the image)
- An oversampling factor to increase the resolution of the image (in this case set to 2)
- The threshold for accepting/rejecting the offset estimate in one window (0.1)

The program generates

- A file containing the list of range/azimuth offsets for the accepted windows (offs, binary)
- A file containing the corresponding SNR values of the correlation (snr, binary)
- A text file containing offsets and SNR values (offsets)

# Refinement of lookup table - Offset computation, considerations

In this example, the quality of the offsets was above the selected threshold for 11 of the 36 estimation windows.

The number of offsets that can satisfy the requirement set by the threshold can differ depending on window size.

Although the program suggests to use the default values for window size and number of windows in the *.diff_par file, it is always recommended to adapt the window size and the number of windows to the specific example being considered.

- If you have many small “topographic elements” use more and smaller windows
- If you have few large-scale feature use less and larger windows
- Use a number of windows that will cover most of the image (i.e., no gaps beween windows)

# Estimation of offsets polynomial

Generation of the polynomial of range and azimuth offsets for the registration of the images

For this we use the program offset_fitm and determine the parameters of a polynomial of the type (a0 + a1 * x + a2 * y + a3 *x * y ….) for the resampling

offset_fitm offs snr 20100813.diff_par coffs coffsets 0.1 1

The program requires

• the files with the offsets estimates and the corresponding SNR value (files offs and snr)

• an indication on the number of terms forming the offset polynomial (in this case 1 → constant offset)

The program will update the offset parameter file by updating the values of the coefficients of the offset polynomial

The program also generates two files (here named coffs and coffsets) containing the culled offsets, i.e. the offsets used for the estimation of the model coefficients, respectively in binary and text form.

# Offset estimation – result

After running the program offset_fitm we can see the result printed on the screen as well as an indication on the quality of the offset estimation

*total number of culling iterations: 2*

*final solution: 10 offset estimates accepted out of 36 samples*

*final range offset poly. coeff.: 0.49311*

*final range offset poly. coeff. errors: 1.43394e+00*

*final azimuth offset poly. coeff.: 0.0727*

*final azimuth offset poly. coeff. errors: 8.71201e-01*

*final model fit std. dev. (samples) range: 0.7047 azimuth: 0.4281*

*NOTE: values might be slightly different because of different software versions*

- The program did 2 iterations to obtain the range and azimuth offset polynomial.
- The program accepted 10 offset estimates after culling (it started with 11 estimates)
- The offset is less than a pixel (both in range and in azimuth) and related errors are small
- The co-registration error between images is of the order of pixel size (due to small number of offsets used)

# Refinement of lookup table - Offset computation (2/2)

Typically an iteration of the offsets estimation is done to refine the offsets polynomial. In this case a larger number of windows (of the same size as in the previous case or smaller are used)

In this case the refinement of offsets and polynomial is done using 32 x 48 = 1536 windows (each 128 x 128 pixels large)

offset_pwrm 20100813.simsar 20100813.mli 20100813.diff_par offs snr 128 128 offsets 2 32 48 0.1

offset_fitm offs snr 20100813.diff_par coffs coffsets 0.1 1

Let's check the results

# Offset estimation – result

After running again the sequence offset_pwrm/offset_fitm we obtain the following results

After offset_pwrm, 856 of 1536 offsets were accepted

Offset_fit required 3 iterations to come to a solution that is slightly better than the previous one.

*total number of culling iterations: 3*

*final solution: 702 offset estimates accepted out of 1536 samples*

*final range offset poly. coeff.: -0.21826*

*final range offset poly. coeff. errors: 1.86039e-01*

*final azimuth offset poly. coeff.: 0.34619*

*final azimuth offset poly. coeff. errors: 6.48501e-02*

*final model fit std. dev. (samples) range: 1.1391 azimuth: 0.3971*

*NOTE: values might be slightly different because of different software versions*

# DIFF&GEO Offset parameter file – final version

```
Gamma DIFF&GEO Processing Parameters
title: DIFF_par determined from ISP image parameter files
initial_range_offset:                0
initial_azimuth_offset:              0
range_samp_1:                     2508
az_samp_1:                        2261
first_nonzero_range_pixel_1:         0
number_of_nonzero_range_pixels_1:   2508
range_pixel_spacing_1:         18.737028
az_pixel_spacing_1:            39.023856
range_samp_2:                     2508
az_samp_2:                        2261
first_nonzero_range_pixel_2:         0
number_of_nonzero_range_pixels_2:   2508
range_pixel_spacing_2:         18.737028
az_pixel_spacing_2:            39.023856
offset_estimation_starting_range:     48
offset_estimation_ending_range:     2460
offset_estimation_range_samples:      32
offset_estimation_range_spacing:      77
offset_estimation_starting_azimuth:   48
offset_estimation_ending_azimuth:   2213
offset_estimation_azimuth_samples:    48
offset_estimation_azimuth_spacing:    46
offset_estimation_patch_width:      128
offset_estimation_patch_height:     128
offset_estimation_threshhold:       0.1
range_offset_polynomial:      -0.21826   0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00
azimuth_offset_polynomial:     0.34619   0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00
starting_azimuth_line:               0
map_azimuth_lines:                   0
map_width:                           0
first_map_range_pixel:               0
number_map_range_pixels:             0
range_looks:                         0
azimuth_looks:                       0
diff_phase_fit:     0.00000   0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00   0.0000e+00
```

# Refinement of lookup table and geocoding

To refine the lookup table based on the offset polynomial use the program gc_map_fine

gc_map_fine 20100813.lt 2976 20100813.diff_par 20100813.lt_fine 1

The new refined lookup table contains now at each pixel (i.e. map position) the correct position of a pixel in the SAR image.

One can check the quality of the refinement by re-transforming the simulated SAR image from the map to the radar geometry with the updated lookup table and displaying it simultaneously to the SAR image. If the 2 images present residual mismatches, the refinement solution needs to be improved (more windows, different size, oversampling, iterative culling etc.)

The command lines are:

geocode 20100813.lt_fine 20100813.EQA.simsar 2976 20100813.simsar.ref 2508 2261

dis2pwr 20100813.simsar.ref 20100813.mli 2508 2508

# Geocoding step

Once the final lookup table has been obtained, it is used to

1) Transform the MLI image from radar to map geometry

NOTE: The same lookup table can be used to geocode any other image that is in the same geometry of the one used to set up the lookup table (e.g., an interferogram where the image acquired on 2010-08-13 is reference)

2) Transform the DEM from map to radar geometry

This transformation will allow processing of differential interferometry. It is however necessary that the interferogram has a master image the one used to set up the lookup table.

# Geocoding the MLI image

To geocode the intensity image we use the program geocode_back

geocode_back 20100813.mli 2508 20100813.lt_fine 20100813.EQA.mli 2976 - 2 0

Input

- The intensity image (and the corresponding width: 2508)
- The refined lookup table

Output

- The geocoded intensity image

Notes

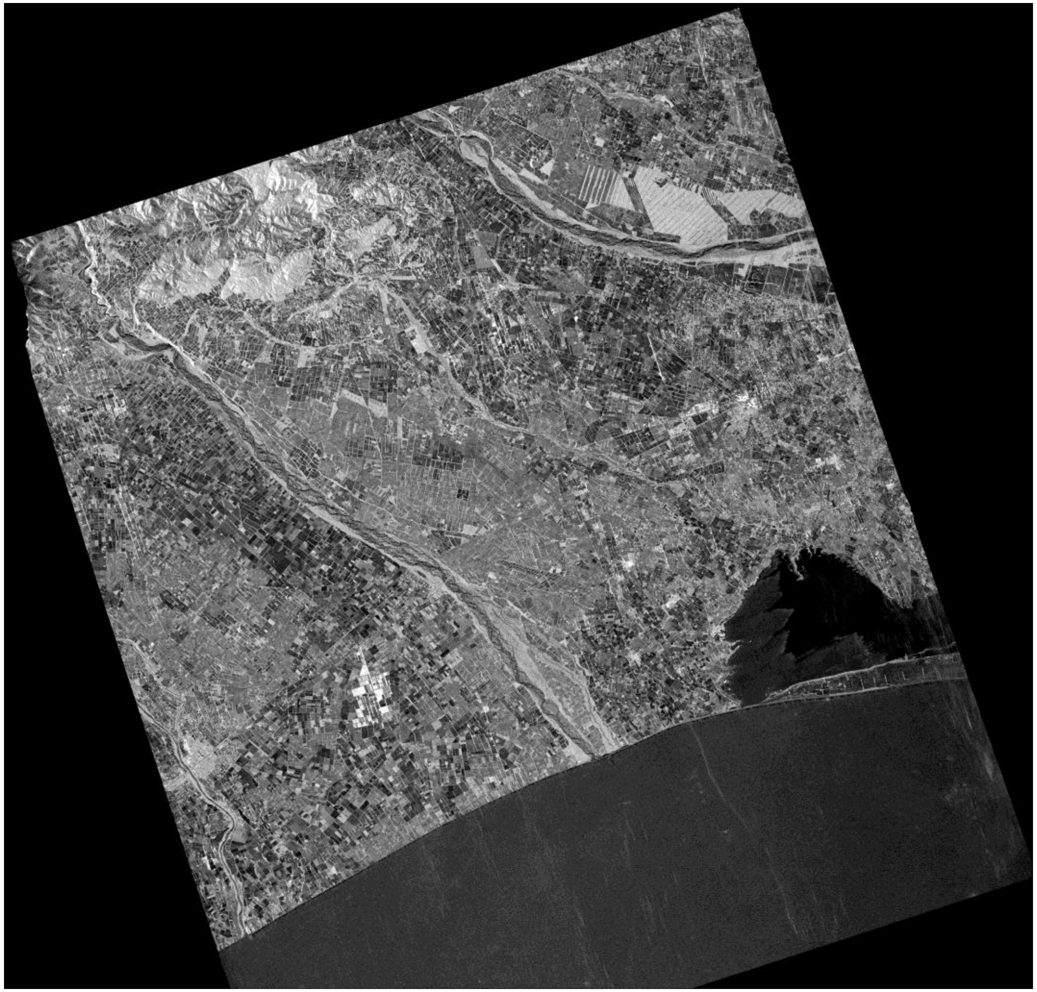
The output width (2976) corresponds to the values reported in the DEM parameter file produced by gc_map (i.e. the file Christchurch.EQA.dem_par)

**This is THE file you use for all information about geocoded products!!!**

# Display of geocoded MLI image

The geocoded MLI image can be displayed with the program dispwr of the DISP module

dispwr 20100813.EQA.mli 2976



*Geocoded ALOS PALSAR image of Christchurch, lat/long projection, pixel size ~ 30 m*

# Transformation of DEM to radar geometry

To transform the DEM from map to radar geometry we use the program geocode

geocode 20100813.lt_fine Christchurch.EQA.dem 2976 20100813.hgt 2508 2261

## Input

- The DEM covering the area of the SAR image (and the corresponding width: 2976) in map geometry
- The refined lookup table

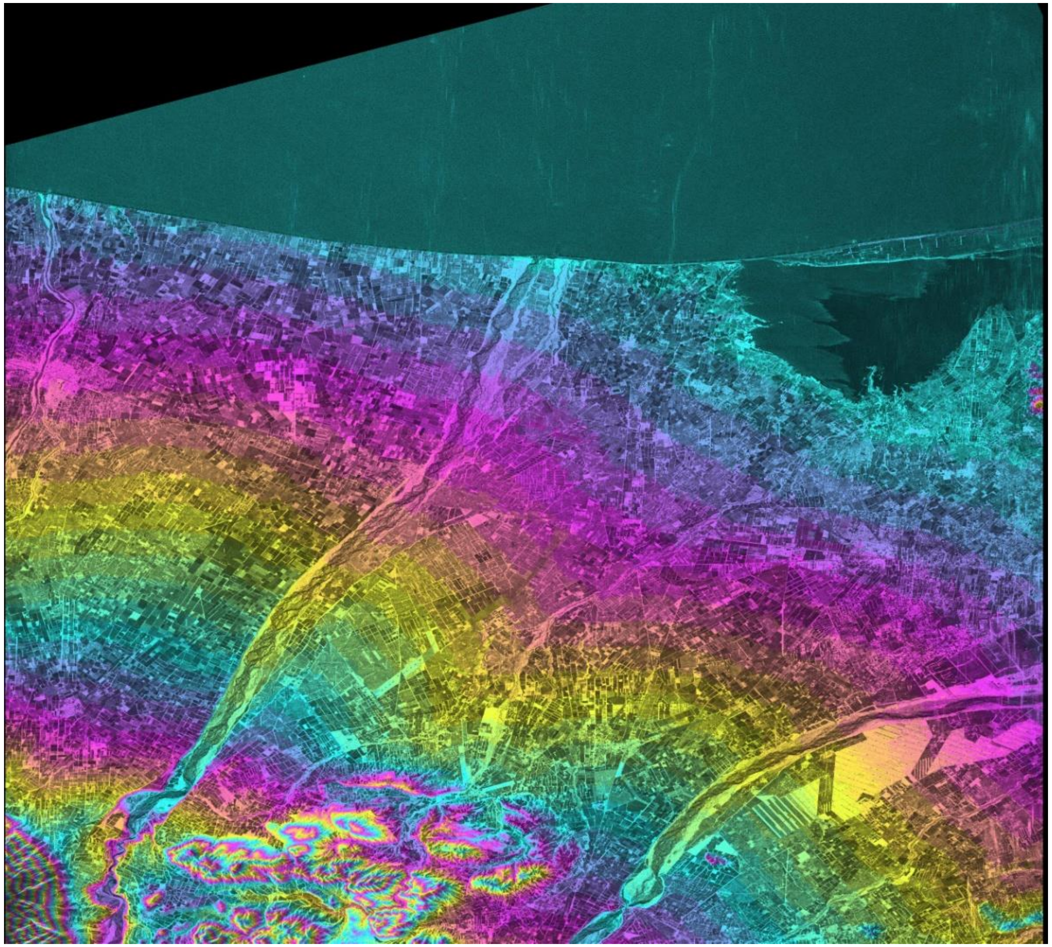
## Output

- The DEM covering the area of the SAR image in radar geometry

# Display of DEM in radar geometry

The DEM in radar geometry can be displayed with the program dishgt of the DISP module. To increase clarity, we display the MLI image as background.

dishgt 20100813.hgt 20100813.mli 2508 - - - 256



*Height map (colour cycle: 256 m) with MLI intensity in background in radar geometry*

# Generation of a kml file and display in Google Earth

To be able to display a geocoded image in Google Earth, the following must be considered

- The image must be in lat/long projection
- The image must be in a format readable by Google Earth

- In our example, the MLI image has been geocoded to lat/long projection.
- With the program raspwr of the DISP module, we can generate a bmp version of the geocoded image. Bmp format is supported by Google Earth.
- Then, we apply the program kml_map to obtain the kml file
- Double click on the kml file will load the MLI into Google Earth for display

raspwr 20100813.EQA.mli 2976 - - - - - - - 20100813.EQA.mli.bmp

kml_map 20100813.EQA.mli.bmp Christchurch.EQA.dem_par 20100813.EQA.mli.kml

# Generation of a geotiff file

To generate a geotiff file from the plain binary file of a geocoded image, use the program data2geotiff

```
data2geotiff Christchurch.EQA.dem_par 20100813.EQA.mli 2 20100813.EQA.mli.tif
```

# Overview

- Meaning of geocoding, DEM
- Image geocoding procedure
- Introduction to the DIFF&GEO module for geocoding
- Geocoding with the DIFF&GEO module
- Summary

# Processing sequence

| **Program name** | **Action** |
|---|---|
| multi_look | Generation of multi-looked intensity (MLI) image from SLC image |
| gc_map | Generation of initial geocoding lookup table. Generation of simulated SAR image |
| geocode | Transformation of simulated SAR image from map to radar geometry |
| create_diff_par | Creation of a parameter file to preserve offset estimates between MLI and simulated SAR image |
| offset_pwrm | Offset estimation between MLI and simulated SAR image |
| offset_fitm | Estimation of model to refine lookup table |
| gc_map_fine | Refinement of lookup table |
| geocode_back | Geocoding of MLI image |
| geocode | Transformation of DEM from map to radar geometry |
| raspwr, kml_map / data2geotiff | Generation of overlay for Google Earth / generate a geotiff image |

# Summary

You have learnt in this section

- What geocoding means
- How geocoding is performed with the GAMMA Software
- How to geocode a MLI image using the DIFF&GEO module
- How to transform a DEM from map to radar geometry
- How to generate an overlay for display in Google Earth
- How to generate a geotiff image