Documentation – User's Guide

Differential Interferometry and Geocoding Software – DIFF&GEO

Geocoding and Image Registration

GAMMA REMOTE SENSING

Version 1.3 – November 2008

GAMMA *Remote Sensing AG, Worbstrasse 225, CH-3073 Gümligen, Switzerland*
tel: +41-31-951 70 05, fax: +41-31-951 70 08, email: gamma@gamma-rs.ch

Table of contents

List of acronyms

ALOS	Advanced Land Observing Satellite
ASAR	Advanced Synthetic Aperture Radar
ASI	Agenzia Spaziale Italiana
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
COSMO-SkyMed	Constellation of Small satellites for Mediterranean basin Observation
DEM	Digital Elevation Model
DIFF&GEO	Differential Interferometry And Geocoding Software
ENVISAT	Environmental Satellite
ERS	European Remote Sensing (Satellite)
ESA	European Space Agency
GEC	Geocoded Ellipsoid Corrected
GTC	Gecoded Terrain Corrected
ISP	Interferometric SAR Processor
JERS	Japanese Earth Resources Satellite
LAT	Land Application Tools
PALSAR	Phased Array L-band Synthetic Aperture Radar
SAR	Synthetic Aperture Radar
SLC	Single Look Complex
SRTM	Shuttle Radar Topography Mission
USGS	US Geological Survey
UTM	Universal Transverse Mercator

1. Introduction

The GAMMA Differential Interferometry and Geocoding Software (DIFF&GEO) is a collection of programs designed to support the SAR Differential Interferometric data processing as well as geocoding between range-Doppler coordinates and map projections. The reason for inclusion of these quite different processes into one software module is that geocoding capability is required for 2-pass differential interferometry. Programs in the DIFF&GEO allow transformation of DEM data into range-Doppler coordinates, transformation of data in radar geometry to map coordinates, precision measurement of offsets between intensity images and calculation of polynomial models of these offsets, simulation of interferometric phase, generating linear combinations of interferograms, stacking of interferograms, generating of a lookup table to map one image into another and resampling of one image into the geometry of another using this lookup table.

In this Manual the programs and the processing steps that support geocoding and image registration with the DIFF&GEO module are described. Programs and processing steps required for differential interferometry can be found on the DIFF&GEO User's Guide on Differemtial Interferometry.

It should be remarked that parameter values provided in the processing examples presented in this document cannot be considered valid for all cases. It is possible that one or more values might have to be adapted to the specific case being processed. It is advised to look carefully at the messages printed on stdout when running each individual program. For assistance please get in contact with us (gamma@gamma-rs.ch).

2. Geocoding theory

Geocoding is the coordinate transformation between the coordinates of an imaging system, in this case range-Doppler coordinates of the SAR, and orthonormal map coordinates.

The position of a SAR image point (pixel) is on a specific circle around the velocity vector of the sensor (intersection of Doppler cone and sphere for constant range). The intersection of this circle with the ground surface together with the knowledge that the SAR is either right- or left-looking determines the target location and the look vector. This is the principle of geocoding.

Geocoding of SAR data means to apply a geometric transformation from the radar geometry to the map geometry. This type of geocoding takes the image in the radar geometry (range-azimuth) and resamples the data into a map projection (UTM, Oblique-Mercator, Polar Stereographic…) with an associated datum. The datum defines the size and shape of the earth and the origin and orientation of the coordinate systems used for mapping. For an example of SAR image geocoding see Figure 1. This type of 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 means to apply a geometric transformation from the map geometry 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. Figure 2 illustrates the transformation of a DEM from map to radar geometry. The DEM in this geometry can then be used for simulating the interferometric phase and differential interferometry.

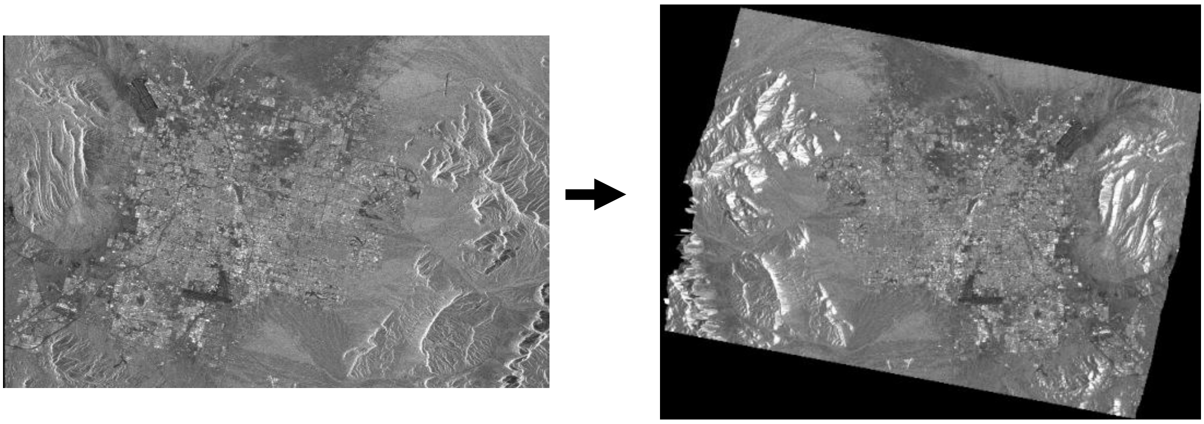


Figure 1. Transformation of image geometry: from radar to map geometry. Example of ERS-1 image acquired over Las Vegas. The urban settlement is recognizable in the centre of the images.

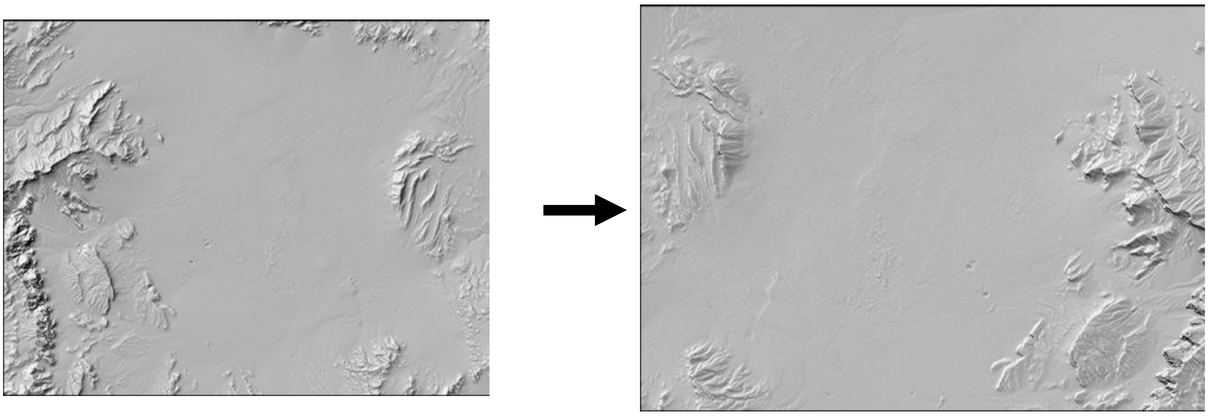


Figure 2. Transformation of image geometry: from map to radar geometry. Example of SRTM-3 DEM of the area of Las Vegas. The DEM has been transformed to match with the SAR image showed in Figure 1.

Depending on the ground surface considered for geocoding, we can distinguish between ellipsoid corrected and terrain corrected geocoding. The corresponding results transformed from SAR to map coordinates are Geocoded Ellipsoid Corrected (GEC) and Geocoded Terrain Corrected (GTC) products. While the generation of GEC products does not require a digital elevation model (DEM), this is necessary for the generation of GTC products. The DEM can either be provided in map projection or in radar geometry (i.e. as derived from interferometry). Surface topography has indeed an effect on the look vector, which means that geocoded products on slope terrain are located at different positions. As an example we illustrate in Figure 3 the result of geocoding a SAR image to the ellipsoid and using a DEM. The two geocoded images are superposed as cyan and red channels respectively. Note the large discrepancies in the mountainous region.

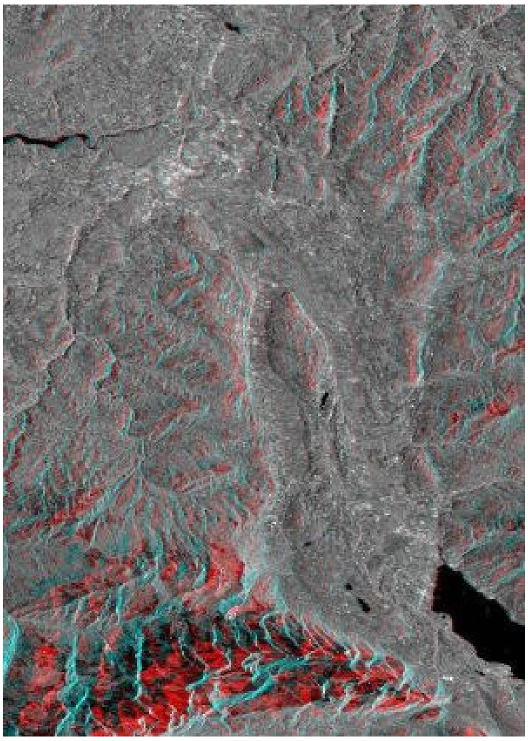


Figure 3. A single scene (ERS-1, orbit 21918) has been geocoded using only the ellipsoid (cyan) and then using the terrain (red) from the RIMINI 250 m DEM over Switzerland. Large discrepancies occur over the mountainous region at the bottom of the image.

3. 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 image data sets from one coordinate system to the other.

The initial determination of the geometric transformation is based on orbital data and ancillary information concerning the image. For GTC geocoding the DEM parameters are also considered. Typically orbital state vectors are affected by errors (RADARSAT, JERS on the order of 1 km, ENVISAT, ERS, PALSAR, TerraSAR-X 1-10m). The initial transformation is therefore not completely accurate because of limitations in the knowledge of the SAR imaging geometry. As a result geocoding with the initial transformation leads to geocoding errors of the order of a few pixels, depending mainly on the accuracy of the orbit data. A geocoding system must be able to correct residual errors due to inaccurate orbit state vectors. It must also be able to correct interpolation errors occurring when the DEM and the SAR image resolution are much different, i.e. several multiples of difference. Finally it has to correct processing system errors (focusing/geolocation due to velocity errors), which can occur in particular in the case of airborne data. To conclude, obtaining a correctly geocoded product requires a refinement of the transformation.

In the refinement step registration offsets are determined between corresponding locations in an original data set and a data set brought into the same coordinate system as the original one.

One way to determine such registration offsets is for example visual interpretation of the geocoded SAR image and a DEM or a topographic map. The user generates a list of pairs of pixel positions for landmarks visible in the reference elevation dataset and in the image to be geocoded (ground control points). The extraction of ground control points by an operator can often be tedious and labor intensive. It also implies the introduction of a source of error due to non-perfect identification of corresponding points.

A more efficient (and probably more accurate) method uses a geocoding lookup table between the two geometries. This procedure is implemented in the geocoding algorithm of the GAMMA software. For each image point defined in one coordinate system the lookup table contains the corresponding coordinates in the other coordinate system. The generation of the look-up table is based on the orbital data and the parameters of the image to be geocoded as well as of the DEM (if GTC considered). If the orbital data are inaccurate, the correspondence in the lookup table contains errors. When geocoding to the ellipsoid, refinement is not possible. This in turn means that the geocoding accuracy of GEC product depends solely on the accuracy of the orbital state vectors. If geocoding with a DEM the refinement of the lookup table makes use of a simulated SAR intensity image based on the DEM and the SAR imaging geometry. The registration offsets between the simulated SAR image and the true SAR image are then determined automatically using cross-correlation analysis.

Either approach (manual control point selection or automated cross-correlation analysis) results in registration offsets, which are then modeled as a bi-linear function of range and azimuth. In this way the transformation gets refined. If working with the geocoding lookup table, the application of the bilinear function to the initial geocoding lookup table results in the refined geocoding lookup table (representing the complete geometric transformation, i.e. initial geocoding and fine registration). The refined geocoding lookup table is then used to geocode data from one coordinate system into the other coordinate system in one step, with the advantage of just a single resampling and interpolation. The initial lookup table is no longer used.

Generation and refinement of the look-up table are further explained in Sections 6 and 7.

Two types of lookup tables are used. The first type of lookup table has the dimension of an image in radar geometry. For each pixel (of the SAR image) the lookup table contains the corresponding map coordinates. In the GAMMA software the corresponding map coordinates are implemented as a pair of real valued numbers (i.e. the real valued row and column numbers of the corresponding target/location in the map projection). The second type of lookup table has the dimension of an image in map coordinates. For each pixel (of the map) the lookup table contains the corresponding SAR range-Doppler coordinates. As in the previous case, in the GAMMA software the corresponding radar coordinates are implemented as a pair of real valued numbers (i.e. the real valued SAR image row and column numbers of the corresponding target/location).

Once a look-up table has been generated, it is possible to operate the geometric transformation of the image from one coordinate system to the other (i.e. radar to map or map to radar). We call the geometric transformation from the coordinates in which the lookup table is given to the coordinates of the lookup table values forward transformation. An example of forward transformation is the geometric transformation of a DEM (or any related product as for example a simulated SAR image from the DEM) from map to radar coordinates given a lookup table in the map coordinates (i.e. containing the corresponding coordinates in the radar geometry). In the forward transformation the input data is of the same dimension (number of columns and rows) as the lookup table. The interpolation necessary to determine the transformed values at the output pixel locations is based on irregularly spaced data. The program ***geocode*** supports this forward transformation (see also Section 7).

We call the geometric transformation from the coordinates of the lookup table values to the coordinates in which the lookup table is given backward transformation. An example of backward transformation is the geometric transformation of a SAR image (or any related product such as an unwrapped phase image or a coherence image) from radar to map coordinates given a lookup table in the map geometry (i.e. containing the corresponding coordinates in the radar geometry). In this case the output data is of the same dimension (number of columns and rows) as the lookup table. The interpolation necessary to determine the transformed values at the output pixel locations is based on regularly spaced data; as a consequence backward geocoding is computationally more efficient than the forward transformation. The backward transformation is generally preferred for its simpler, and therefore better controlled and more efficient, resampling. The program ***geocode_back*** supports this backward transformation (see also Section 9).

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. In resampling SAR images from a higher resolution to a lower resolution the signal noise level is kept unnecessarily high. It is better to first apply multi-looking (or filtering) to reduce the spatial resolution to a similar value as required for the resampled output image. To select an output spatial resolution much higher than the input spatial resolution strongly reduces the geocoding efficiency without much information gain. In such a case the geocoding can be conducted to a lower resolution followed by an oversampling to the required higher resolution.

Due to the special SAR imaging geometry effects such as layover and shadowing occur. In the case of layover the ground surface interacts more than once with the circle of possible target locations, i.e. several locations contribute to the signal. There are various possibilities to deal with this problem. With the GAMMA software the users has the possibility to select one of the following options: (a) set areas of layover and shadow as missing values; (b) keep the actual value, i.e. assuming the same SAR values to different positions of the map; (c) interpolate across regions of layover and shadow.

The typical geocoding processing sequence is summarized in the flow chart given in Figure 4. Details to each processing step are provided in the following Sections as reported below. If a DEM is available, at first a DEM parameter file has to be generated, in which the DEM characteristics, such as size, posting, projection, datum etc., are reported. Pre-processing of a DEM becomes necessary if any of the parameters of the DEM should be changed (e.g. projection, size, pixel spacing etc.). Details are provided in Examples A and B. Initial determination of the transformation with the generation of the lookup table is described in Section 6. Additional products such as the local incidence angle image, the pixel area normalization image etc. can be generated. The procedure for the refinement of the look-up table is described in Section 8.

Depending on the geometry of the look-up table input and output of forward and backward geocoding are different. Section 8 provides more details on this topic.

1) A SAR to MAP lookup table has the same dimension of the map image and contains the coordinates of the radar geometry. Forward transformation means to transform an image from map to radar coordinates. Backwards transformation means to transform an image from radar to map coordinates. This type of lookup table is used for the geocoding of a SAR image or for the transformation of a DEM to radar geometry.
2) A MAP to SAR lookup table has the same dimension of the image in radar geometry and contains the coordinates of the map geometry. Forward transformation means to transform an image from radar to map coordinates. Backwards transformation means to transform an image from map to radar coordinates. Typically this is the case when the InSAR height image in radar geometry is used for geocoding.

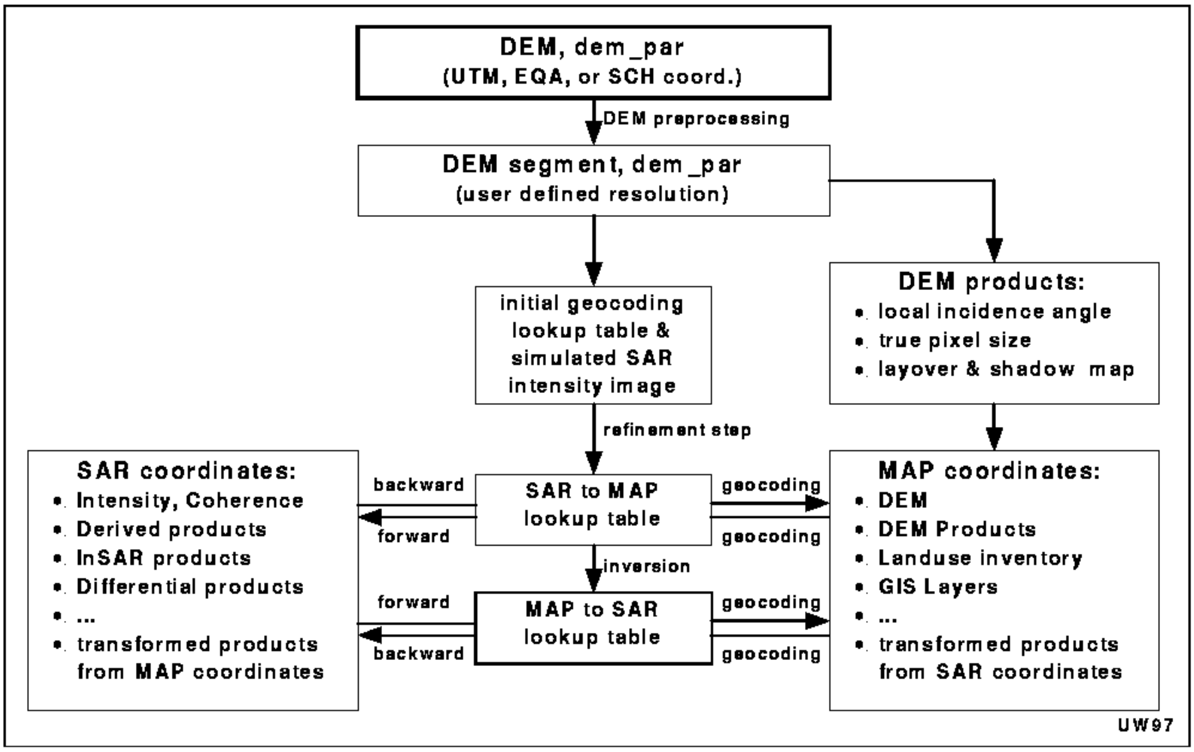


Figure 4. Geocoding processing sequence

4. Generation of DEM parameter file

If the geocoding encompasses the use of a DEM, the processing starts with the preparation of the DEM and the generation of the corresponding DEM parameter file.

The DEM has to be a pure binary file (no header). Supported formats are (4-byte) floating point and (2-byte/short) integer. The byte order has to be big endian. If the DEM has been obtained with MS-Windows-based software, the byte order is little endian. To make the DEM compatible with the GAMMA software programs, byte swapping is necessary. This operation is supported by the program ***swap_bytes*** (in the DISP package), which will generate a new DEM file. If you are unsure about the byte order of the DEM, you can display it with the program ***dishgt***. If the byte order is incorrect, the elevation values displayed are completely wrong.

Independently whether geocoding makes use of a DEM or not, the first step is to generate the **DEM/MAP parameter file.** This ASCII-file contains all relevant information about the MAP projection used for geocoding. In other words it describes the geometry of an image in map coordinates.

The structure of the DEM/MAP file consists of 4 parts
1) General parameters
2) Ellipsoid parameters
3) Datum parameters
4) Projection parameters

The exact definitions and formats of the individual parameters are specified in the Reference Manual under DEM_par. Information about supported projections, ellipsoids and datums can be found in the Reference Manual in the Section "File Formats and Auxiliary Files".

To generate DEM/MAP parameter file the program ***create_dem_par*** must be used. The program runs interactively. In Example A we demonstrate how to generate a DEM/MAP parameter file for a SRTM DEM file. If a DEM is not available, the user can still create a DEM/MAP parameter file specifying a constant height value. In other words this is the starting point for the generation of Geocoded Ellipsoid Corrected (GEC) products. In this case the user can aid the program in finding the corner coordinate, i.e. the area covered by the SAR image to geocode by adding the ISP parameter file at the command line.

Tip: It is suggested to use a file name of the type *.dem for the DEM image file and *.dem_par for the corresponding DEM/MAP parameter file.

The user can edit the DEM/MAP parameter file with a text editor.

To display a DEM and assess its quality a shaded relief can be generated. To display a DEM as shaded relief use the program ***disshd***. If the pixel geographical coordinates shall be displayed as well, use the program ***disdem_par*** instead. To generate a SUNraster or bmp version of the DEM as shaded relief use the program ***rasshd***. The example in Figure 5 shows the UTM DEM (shaded relief) for Death Valley.

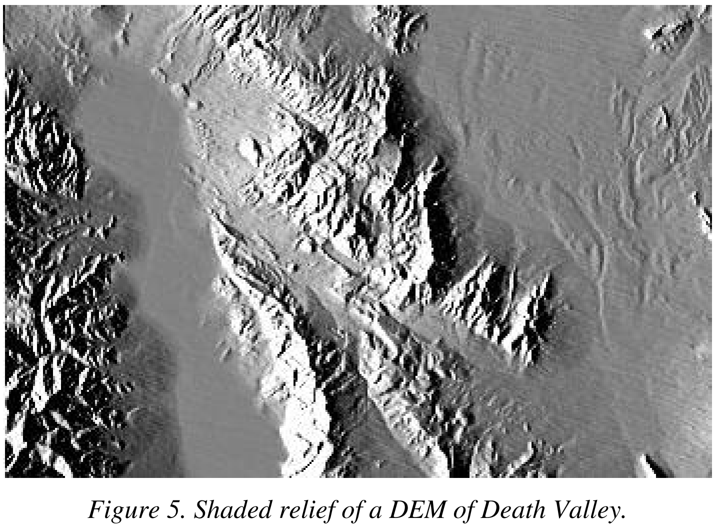


Figure 5. Shaded relief of a DEM of Death Valley.

If the DEM presents (small) gaps, it is possible to fill them by using 2-D interpolation. For this the DIFF&GEO module offers the program ***interp_real***. A more advanced solution is implemented is the ISP program ***interp_ad*** which performs a weighted interpolation of gaps in 2D data using an adaptive smoothing window. The user can specify the radius of the window, the minimum and maximum number of samples to be used for the interpolation as well as the interpolation algorithm and the weighting function. This operation is further illustrated in Example B.

5. DEM pre-processing

If projection, size, location of the DEM to be used for geocoding is not the final one, it is possible to transform the DEM to the desired geometry with the program ***dem_trans***. The program interpolates the original DEM to the new geometry. As an example let us consider the freely available SRTM-3 DEM. Data files are provided in the Equiangular projection with a pixel spacing of approximately 90 m in both directions. Geocoding to a higher spatial resolution and/or to another projection can be then only performed if the DEM is transformed according to the final map geometry. Pre-processing of SRTM-3 DEM data is presented in more detail in Examples A and B.

6. Generation of the geocoding look-up table

The initial geocoding lookup table is calculated based on heights in MAP coordinates and parameters describing the SAR imaging geometry. The lookup table is complex valued. For each pixel the lookup table contains a complex number (FCOMPLEX format) with the real part corresponding to the x (resp. column / range / easting / longitude) coordinate and the imaginary part corresponding to the y (resp. row / azimuth / northing / latitude) coordinate.

The dimension of the look-up table corresponds to the dimension of the reference image used for geocoding. In other words if we are geocoding to map coordinates, the look-up table will have the dimensions of the reference image in map coordinates (DEM or ellipsoid). If the reference is an interferometric height map, the look-up table will have the dimensions of this image. Each complex number is represented by two real valued numbers (as compared to integer numbers) in order to optimize the geocoding accuracy. Figure 6 illustrates the look-up table for the transformation between map geometry and radar geometry. The size of the look-up table coincides with the size of the DEM used for geocoding. Each pixel contains the corresponding coordinates in the radar geometry.

Below we illustrate how to generate a look-up table depending whether a DEM in map coordinates (or just the ellipsoid) or a height map in radar geometry is provided.

6.1. Look-up table based on DEM / ellipsoid in map geometry

To generate a lookup table for GTC products (i.e. DEM available) either the program ***gc_map*** or ***gc_map_grd*** must be used, depending whether the SAR image is in slant range or ground range geometry.

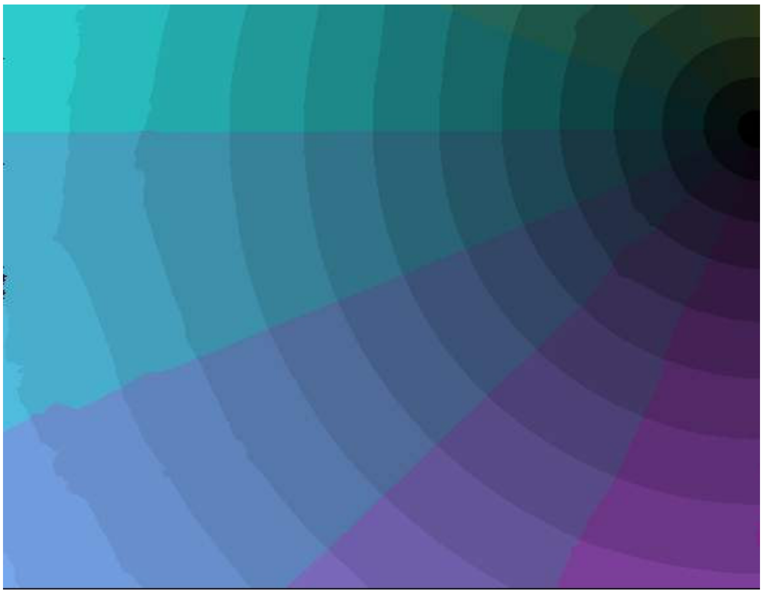


Figure 6. Example of look-up table. Given a DEM in UTM projection being 2376 pixel wide and 1848 pixels long, we generated with gc_map the look-up table having the same dimensions as the DEM. The complex-values look-up table contains at each position the corresponding coordinates in the range-azimuth geometry of the SAR image to be geocoded. It should be noticed that what is here illustrated represents the magnitude of the complex values. Errors in the look-up table would show up as inhomogeneities in the figure display (jumps, zero values etc.)

To generate a lookup table for GEC products (i.e. no DEM available) either the program ***gec_map*** or ***gec_map_grd*** must be used, depending whether the SAR image is in slant range or ground range geometry.

The DEM map geometry and the SAR range-Doppler imaging geometry are used to determine the initial transformation lookup table. For each pixel of the DEM (here we refer to the ellipsoid as well) the DEM coordinate is first transformed from the map coordinates to Cartesian coordinates. Next the Datum shift between the Datum of the reference ellipsoid used for the DEM map projection and the reference ellipsoid used for the description of the sensor orbit geometry, are corrected for. Then the acquisition time and position corresponding to the selected map coordinate are determined. Finally, the look vector to the selected map coordinate is calculated leading to the desired azimuth and slant range position. Image skew (for non-deskewed images) and the local terrain height are taken into account in these transformations. After the above transformations the SAR look vector and the terrain surface coordinates are available in the same Cartesian coordinate system.

When doing GTC, it is possible to generate from the DEM a simulated SAR intensity image. The simulated SAR image is useful in the process of refining the look-up table (see Section 8). The backscattering of the simulated SAR image is found by multiplication of the calculated pixel area with an empirical function of the local incidence angle which is used to account for the incidence angle dependence of the backscattering coefficient. The image simulation does not consider the dependence of the backscattering on the surface type. As a result an almost constant value is obtained for flat areas. Figure 7 for an example of the

Landers area, California. Topographic features appear as in a real SAR image, stretched along the direction of the image acquisition. The radar is in this example on the left side of the image looking out to the right.

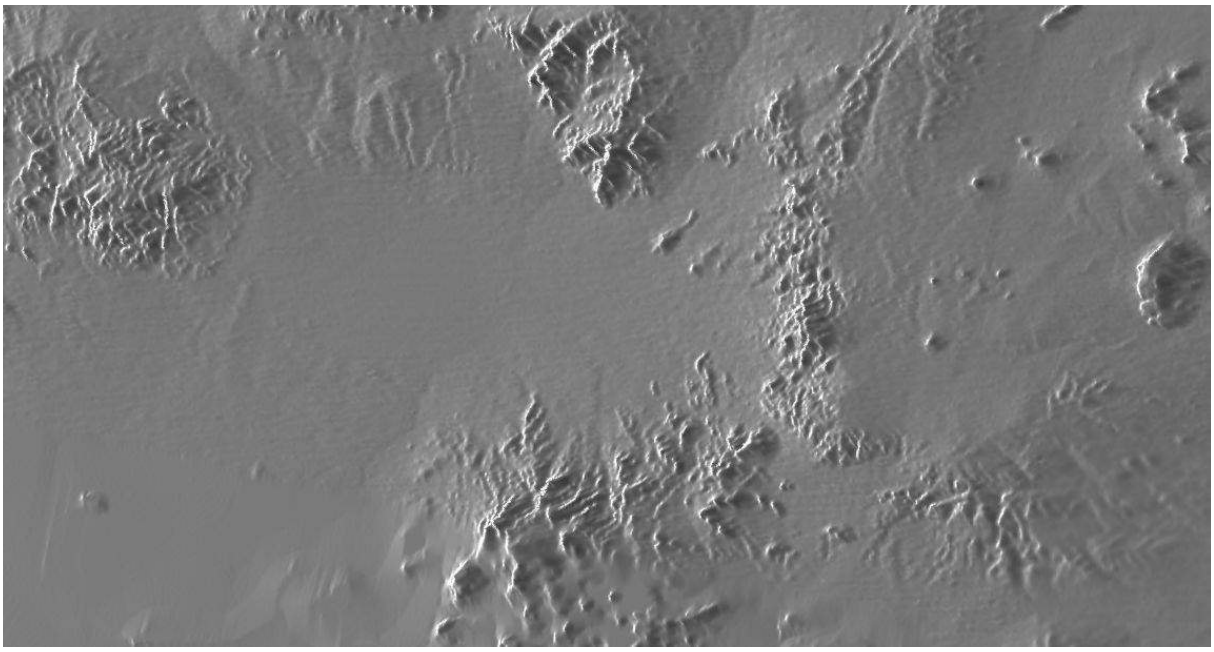


Figure 7. Simulated SAR image from a DEM of the Landers area, California.

To further highlight the properties of the simulated SAR image, Figure 8 shows an example of a simulated SAR image for a part of Death Valley, USA. The valley floor, in the center of the image, is flat. The simulated backscattering is almost constant. The real SAR image shows a lot of backscatter variation for the same area because of the different surface types. For the sloped areas, on the other hand, the local topography has a strong influence on the backscatter.

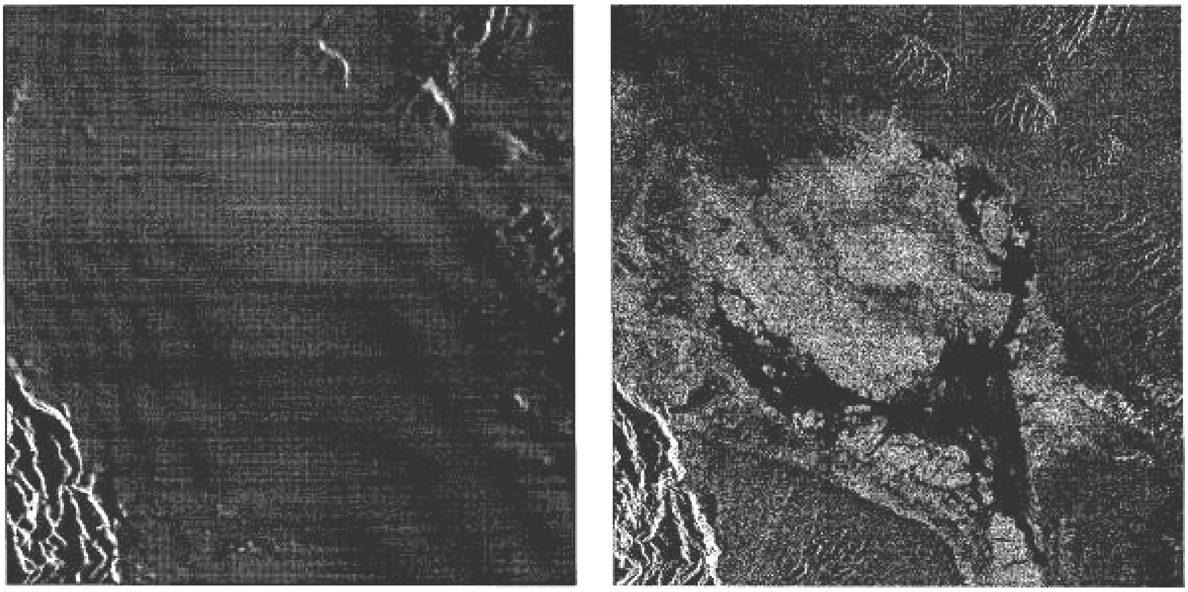


Figure 8. Simulated SAR intensity image (left) and actual SAR intensity image (right) of Death Valley (Wegmüller et al., 1999). The simulated SAR image is here in the radar geometry. This has been obtained by transforming it to the same coordinate system of the SAR image.

For GTC geocoding, i.e. if the DEM in coordinates is available, it is also possible to directly calculate the following geometric parameters

- Spherical angles of local surface normal (slope and aspect angle)
- Local incidence angle
- Projection angle
- Pixel size normalization factor
- Layover and shadow map

The imaging geometry with the different angles and vectors used is shown below.

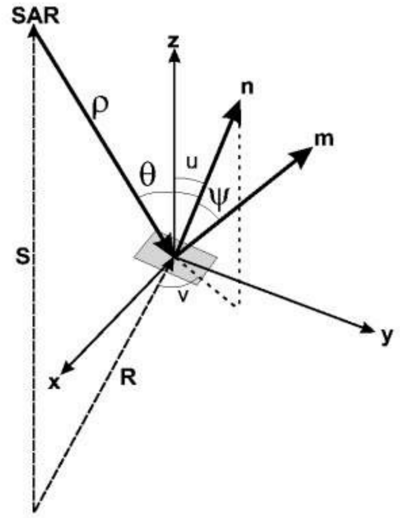


Figure 9. Geometric parameters of a SAR observing system.

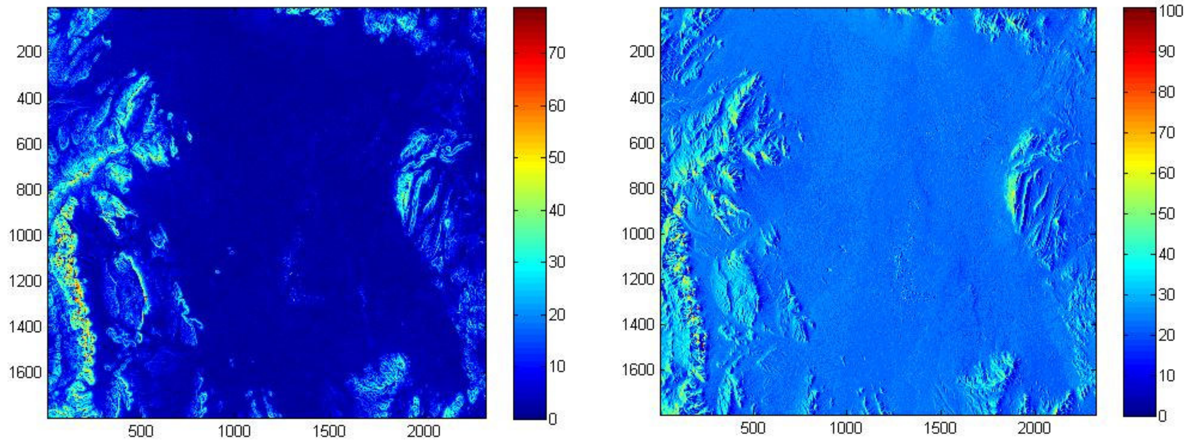
The local incidence angle, θ, is defined as the angle between the local surface normal, n, and the negative of the radar look vector ρ. The projection angle, ψ, is defined as the angle between the local surface normal, n, and the image plane normal, m, the latter defined by the vectorial product (ρ x v) with v the sensor velocity vector. As an effect of surface topography the true pixel size, that is the area contributing to the signal represented in one pixel, is not constant. The pixel size normalization factor is the factor used to normalize the backscattering for the terrain induced variation of the pixel size. The spherical angles u and v of the local surface normal describe the orientation of the local normal vector.

All images but the layover and shadow mask are in floating point format. The information in the images of slope angle, aspect angle, local incidence angle and projection angle is given in radians. The pixel size normalization factor is dimensionless. Figure 10 shows an example of color-coded geometric parameters for the DEM illustrated in Figure 2 and the ERS-1 image in Figure 1 (23 degrees incidence angle, right looking, descending pass). Angles have been expressed in degrees for clarity.

The layover and shadow mask is in byte format. The table below indicates which values are used for which effect. We distinguish between true layover and layover as well as true shadow and shadow; an explanation is given in Figure 11. A layover/shadow map for the DEM and the ERS image considered in Figure 1 is shown in Figure 12. Layover appears in areas of strong topography (light blue). Because of the steep look angle, no shadow has occurred. The layover and shadow map.

Table 1. Explanation of layover and shadow mask image values.

Value	Effect	Description
0	NOT_TESTED	No effect
1	TESTED	Neither layover nor shadow
2	TRUE_LAYOVER	Pixel were slope angle is greater than look angle (see Figure 11)
4	LAYOVER	Pixel in area affected by layover (see Figure 11)
8	TRUE_SHADOW	Pixel were opposite of slope angle is greater than look angle (see Figure 11)
16	SHADOW	Pixel in area affected by shadow (see Figure 11)



(a) *(b)*

Figure 10. (a) Local slope angle and (b) local incidence angle for a DEM of the area of Las Vegas and an ERS-1 image. The centre of the area is rather flat showing slopes of approximately 0 degrees and local incidence angles of approximately 23 degrees.

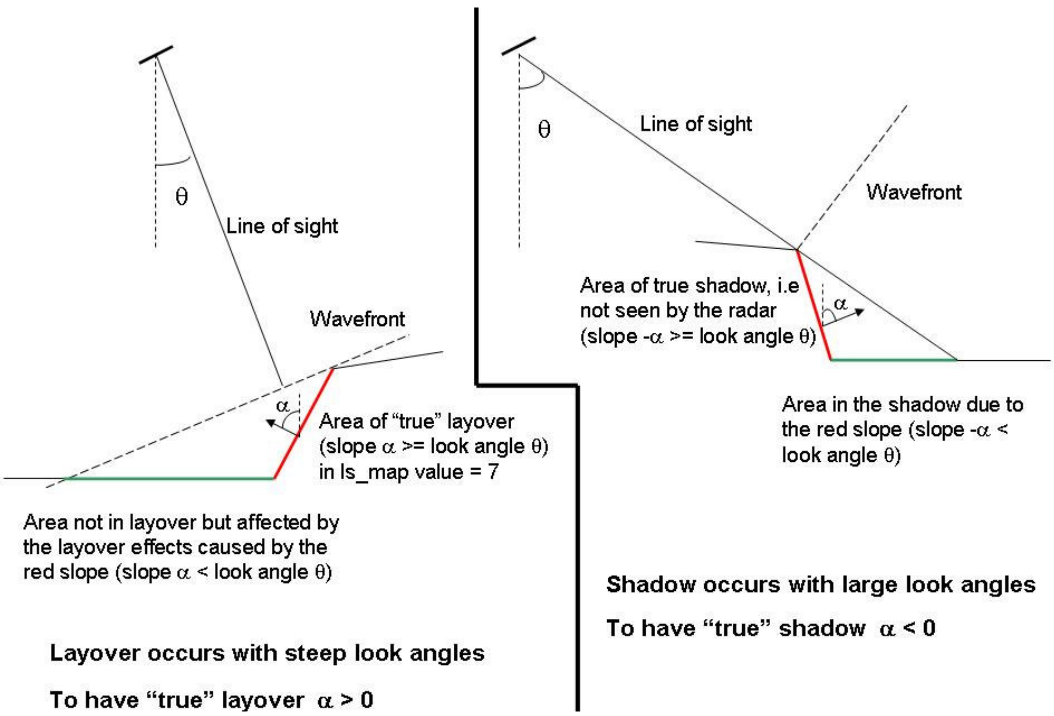


Figure 11. Explanation of layover and true layover (left side of the Figure), shadow and true shadow (right side of the Figure).

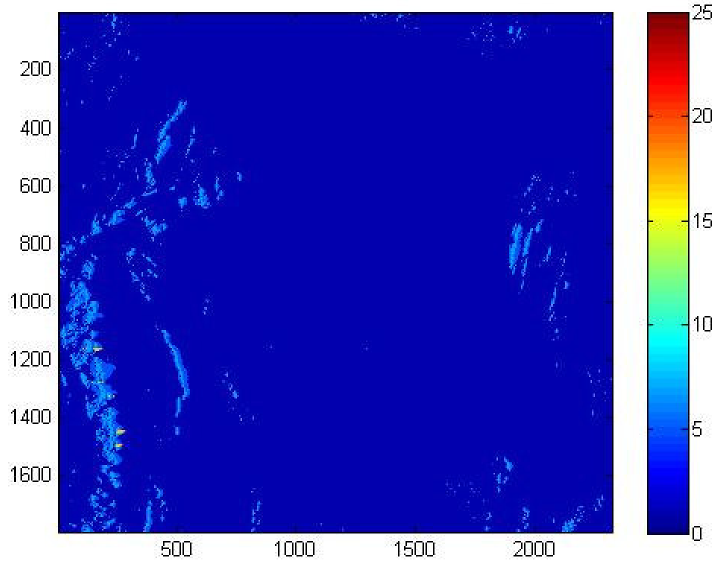


Figure 12. Layover/shadow map for the DEM / ERS-1 image combination described in Figures 1 and 2.

6.2. Look-up table based on InSAR-derived DEM in radar geometry

To generate a lookup table from an InSAR-derived DEM in radar geometry and available map projection use the program ***gc_insar***. The program requires in input the ISP SLC parameter file of the reference image, the ISP offset parameter file describing the interferogram, the height map obtained from the unwrapped flattened interferogram (see User's Guide of the ISP module for more details) and a DEM parameter file describing the map projection to which images shall be geocoded. The output is represented by the lookup table in radar geometry (containing corresponding map coordinates)

7. Forward transformation using the program geocode

Once a look-up table has been obtained, it is possible to transform an image given in the coordinate system in which the look-up table is expressed to the geometry contained in the look-up table. Two applications are therefore possible

- DEM or DEM product (e.g. the simulated SAR image) in map geometry and look-up table in map geometry. With the program ***geocode*** the forward transformation that converts the DEM or DEM product in the radar geometry is applied. The simulated SAR image in radar geometry can be used for refining the look-up table (see Section 8).
- InSAR height map or any SAR image matching with the InSAR height map and look-up table in radar geometry. With the program ***geocode*** the forward transformation that converts the InSAR height map or any SAR product to the map geometry is applied.

Considering the transformation is based on an irregularly spaced data sample, for the forward transformation nearest neighbor or various other interpolation algorithms can be selected.

With the program ***geocode*** the forward transformation of

- real valued data (4 byte float format, examples are intensity, coherence map, height, etc.),
- complex valued data (pairs of 4 byte float format, examples are complex interferograms and complex differential interferograms),
- data of unsigned char format (examples are the phase unwrapping flag file and the layover and shadow map), and
- image data in SUNraster format (examples are visualization products, classification results)

is possible.

Figure 2 showed an example of forward transformation: the DEM in map geometry obtained with ***gc_map*** has been transformed to radar geometry using the program ***geocode***. The DEM is 2376 pixels wide and 1848 pixels long with pixel spacing of 25 m in both directions. Forward transformation generated a DEM in radar geometry being 2500 pixels wide and 1800 pixels long with 20 m pixel spacing.

8. Refinement of geocoding look-up table

When geocoding a SAR image and a DEM is available (i.e. GTC geocoding), the look-up table should be refined to compensate for location errors due to imprecise orbital parameters. The refinement consists of the following steps further explained in following sub-Sections:

1. Computation of offsets between simulated SAR and original SAR image
2. Estimation of polynomial for transformation between the two geometries
3. Refinement of the look-up table using the correction polynomial

To appreciate the geocoding error between if an initial look-up table is used, we can compare the simulated SAR image transformed to radar geometry with the look-up table and the actual SAR image. Typically the two images will present a small offset, particularly along the range direction. This reflects the imprecision of the orbital data on which the construction of the correspondence between map and radar geometry was based. The consequence is that if the SAR image were geocoded with this look-up table, it would not match with the reference map geometry.

To improve the relationship describing the correspondence between the map and the radar geometry, we construct a model that describes the offsets between the simulated SAR and the real radar geometry. For the model one can choose whether a simple constant or a linear polynomial is used. By applying this model to the look-up table, the location of each pixel within the look-up table will be corrected leading to the correct correspondence between map and radar geometry.

8.1. Offset computation

The procedure to compute the offsets between the simulated SAR image (in the radar geometry) and the actual SAR image is similar to the procedure described in the User's Guide of the ISP module to compute the offsets between two SLCs forming an interferometric image pair.

At first a parameter file containing offset information, as well as information on the two images being compared needs to be generated. This is done with the program ***create_diff_par***. A DIFF/GEO parameter file will be generated. The program is interactive in the sense that the user is required to generate at the command line the file itself. If specific values are not known a priori, the user can accept the default values being suggested. With ***create_diff_par*** the file is just initialized. It will be updated throughout the next processing steps and will contain the offset information.

To obtain an idea of the approximate offset the program ***init_offsetm*** can be used. This program generates a simple offset in range and in azimuth based on the cross-correlation algorithm applied to a subset of the entire image.

To improve the estimates of the offsets, accurate offsets are determined for a number of image chips, in each of which the cross-correlation algorithm is applied. The algorithm is implemented in the program ***offset_pwrm***. Size and number of the image chips depends on DEM and SAR image resolution as well as on the features being observed in the two images. The size of the search window shall be adapted to the resolutions of SAR data and DEM. For SAR/DEM pixel spacing on the order of 25 m, a search window size of 128 or even 256 is appropriate. For higher resolutions (e.g. 100 m) a search window size of 64 is sufficient. If the area presents strong topographic features, with high frequency, a large number of image chips (e.g. 48 or 64) with small windows (e.g. 128 or 256) will provide more accurate results because the correlation is enhanced. On the contrary when working with images of rather flat areas it is more convenient to set the image size to a larger value (e.g. 512) and use fewer windows (e.g. 16, 24 or 32). It is suggested that, prior to the offset computation, the images are displayed in order to obtain an idea on how large an image chip shall be and, consequently, how many are needed to cover the image uniformly.

An important parameter in offset estimation is the signal-to-noise ratio (SNR) expressing the reliability of the correlation estimated in a specific window. Typically for image chips that show SNR values below 7.0, the offset estimate is likely to be incorrect. Setting a threshold to the SNR for example at 7.0 has the effect that only offset estimates corresponding to images with a significant image contrast are kept. In this way we want to ensure that the estimation of the coefficients in the polynomial that will be used for correcting the look-up table will not be affected by spurious offset estimates. If only very few offsets are being accepted, it might be necessary to use a slightly lower threshold. It should however kept in mind that this action might lead to some imprecision in the refinement. This case is encountered when topography is very flat or when dealing with shallow look angles (e.g. JERS-1 and PALSAR) and moderate topography.

8.2. Estimation of fine registration offset polynomial

Once a grid of offset estimates has been obtained, the coefficients of the polynomial that describe the correspondence between the actual radar image geometry and the (incorrect) radar geometry of the simulated SAR image can be computed. The offsets are ingested to a least-squares regression procedure implemented in the program ***offset_fitm***. The user can specify the number of coefficients that shall be used in the model: 1, 3 and 4. If 1 is chosen the offset polynomial will corresponds to a simple constant shift in range and in azimuth. If 3 and 4 are chosen, linear dependence of the offsets with respect to the range and the azimuth direction are considered. A polynomial order parameter of 3 is usually appropriate in the case of geocoding refinement. By choosing 4 the user considers also a coupled effect of the range and azimuth direction. This solution is suggested when dealing with long stripes of data to be geocoded.

The sequence ***offset_pwrm / offset_fitm*** can be repeated a couple of time to improve the quality of the estimate, because the bilinear co-registration function is used to guide the search of the positions where cross-correlation is applied. In a first step, a small number of offsets and an oversampling factor of one can be used with high efficiency. Then, a large number of offsets and oversampling factors of 2 or 4 can be used to improve the quality of offsets and consequently of bi-linear polynomial.

If the result of the automatic offset estimation is not satisfactory the user can manually pick ground control points in the two images, the offsets between which will be then used in the estimation of the polynomial for the refinement of the lookup table.

The program that allows selecting ground control point is ***gcp_2ras***. This program reads and displays 2 images in SUNraster or bmp format. The user can then select corresponding points in the two images and save the coordinates in a text format file. This file can be edited using a text editor if necessary. It is suggested to select a sufficient number of points over the entire image to be able to generate a reliable offset polynomial (at least 10-20). It is also suggested that these points be selected in areas of relatively flat terrain using features such as roads, and rivers as matching features.

To determine then the offset polynomial use the program ***offset_list_fitm***. Similarly to the ***offset_fitm*** program the offsets stored in the list of GCPs are used to estimate the coefficients of the polynomial for the refinement of the lookup table.

8.3. Refinement of look-up table

The fine registration polynomial is finally used to refine the look-up table. For this use the program ***gc_map_fine***. The refined look-up table points directly from the map geometry into the radar geometry. Once the refined lookup table is available, the less accurate initial lookup table is no longer used.

To appreciate whether the new lookup table is more precise, the simulated SAR image in map geometry can be transformed to the radar geometry with the refined lookup table using the program ***geocode***. If still some mismatches should occur, the refinement procedure described in this Section can be repeated.

9. Final geocoding

Once the refined lookup table is available, backward or forward geocoding can be applied.

9.1. Backward geocoding

With backward geocoding a SAR image (or an interferometric product) is transformed from radar to map geometry. More in general this is the geometric transformation from the coordinates of the lookup table values to the coordinates in which the lookup table is given. In this case the output data is of the same dimension (number of columns and rows) as the lookup table. The backward transformation is supported by the program ***geocode_back***. The interpolation necessary to determine the transformed values at the output pixel locations is based on regularly spaced data. Therefore, backward transformation is much faster than forward transformation. Nearest neighbor, spline, spline-log, bilinear, and bilinear-log interpolation algorithms can be selected for data resampling.

Backward transformation is supported for real valued data (4 byte float format, examples are intensity, coherence map, height, etc.), complex valued data (pairs of 4 byte float format, examples are complex interferograms and complex differential interferograms), data of unsigned char format (examples are the phase unwrapping flag file and the layover and shadow map), and image data in SUNraster file format (examples are visualization products, classification results).

The geocoded SAR image or any interferometric product can be compared to the DEM in map coordinates or any other geometric product obtained with ***gc_map***. Figure 1 illustrates the result of terrain geocoding from radar to map coordinates.

9.2. Forward geocoding

With forward geocoding the DEM or any DEM product is transformed from map to radar geometry. This has been described in Section 7. Compared to any product obtained when running ***geocode*** with the initial lookup table, geocoded products in radar geometry based on the refined lookup table will be more precisely located.

Since forward geocoding is based on interpolation of irregularly spaced data while backward geocoding is based on interpolation of regularly spaced data, forward geocoding is less efficient than backward geocoding, i.e. the program geocode takes more time than the program geocode_back. To make use of a backward transformation in the process of forward transformation, the lookup table can be inverted, i.e. the input geometry becomes the output and vice versa. The combination of lookup table inversion and backward transformation corresponds to a (more efficient) forward transformation.

Nearest neighbor resampling (mandatory for SUNraster file and unsigned char formats) may perform better using the inverted lookup table because the resampling and interpolation necessary in the lookup table inversion may have removed noise.

To invert the geocoding lookup table use the program ***gc_map_inversion***. The "inverted" lookup table has the dimension of the target coordinates of the input lookup table. If a DEM in MAP coordinates is used to generate a lookup table *.MAP_to_SAR then the inverted lookup table SAR_to_UTM has the dimension of the SAR data files with each complex value containing the corresponding map coordinates.

The inverted lookup table contains for each pixel of the SAR image (range and azimuth coordinates) a pair of real values corresponding to the map coordinates (characterized in the

DEM/MAP parameter file). With this lookup table images in the initial reference geometry can be transformed to the output geometry (e.g. from map to radar geometry if lookup table in map geometry) using the program ***geocode_back***.

10. Additional geocoding tools

In this Section we present a set of tools that complement the geocoding procedure described so far. Tools include the possibility to compute the coordinate in map geometry given the coordinate in the radar geometry or vice versa for one or a list of points, transform an image from one map projection to another, calculate the terrain normal or gradient vector of a DEM, and calculate the SAR look-vector direction in map geometry.

10.1. Coordinate transformation tools

To calculate SAR pixel coordinates from geographic coordinates use the program ***coord_to_sarpix***. If no DEM parameter file is specified the program assumes that equiangular coordinates are provided, i.e. latitude and longitude. The radar geometry is specified by the ISP parameter file.

To calculate geographic coordinates of a SAR pixel and image corners use the program ***sarpix_coord***. If no DEM parameter file is specified the program will generate the geographical coordinates as latitude and longitude.

10.2. Map projection transformation tools

When geocoding a SAR image the user specifies a map projection. It can however happen that for one or more pixels the coordinates in another map projection is required or that the entire image shall be transformed to another map projection for further utilization.

For example a SAR image might have been geocoded to the Albers Conical Equal Area projection and we want to know the coordinates of the image corners as latitude and longitude to identify the area covered by the image. For this we can use the program ***coord_trans***. The program runs interactively and allows obtaining for a single coordinate the corresponding value in one of the projections supported by the GAMMA software.

To transform an image from a map coordinate system to another, use the program ***map_trans***. This program is useful if two images in different map projections need to be compared. The Examples section of this document includes processing examples of multi-source image registration. The program ***map_trans*** requires the input image, the corresponding DEM parameter file and the DEM parameter file of the output projection. The output image will be fully described by this DEM parameter file. The output DEM parameter file can be generated with the program ***create_dem_par***.

10.3. Computation of DEM normal / gradient direction

In some applications, as for example surface deformation monitoring, information on the normal to the DEM surface and the DEM surface gradient can be very helpful. The surface gradient vector is defined as the projection of the surface normal on the surface plane which corresponds to the local direction of steepest descent. Since differential interferometry provides the deformation component in the look vector direction, the DEM gradient expressed in map geometry helps in the interpretation of surface deformation.

To calculate the terrain height normal or gradient vector direction in map geometry use the program ***dem_gradient***. Based on the DEM geometry the terrain height normal or gradient vector is calculated for each pixel of the DEM/map section specified in the DEM parameter file. The local topography is taken into account. The direction vector is expressed in the local map coordinates through the elevation angle and the orientation angle, both calculated at each map pixel. For the computation the DEM/map parameter file is required. The output direction (surface normal or gradient) vector elevation and orientation angles are stored in separate binary files (float).

10.4. Computation of look direction

Another tool that is useful in surface deformation monitoring based on differential interferometry is the program ***look_vector***. This program determines the look vector direction in terms of elevation (theta) and orientation (phi) angle fields (in map geometry). The look vector is assumed to point towards the radar. The look vector information as provided by the program ***look_vector*** is used in the calculation of 3D look vector fields from differential interferometric observations and a deformation direction field as supported by the programs ***dismap_vec*** and ***dispmap_vec2*** (see DIFF&GEO User's Guide on Differential Interferometry and the DIFF&GEO Reference Manual)

11. Co-registration of two SLCs using a lookup table

In the User's Guide of the ISP module a procedure for co-registering two SLC images based on the cross-correlation algorithm has been presented. Several image patches uniformly distributed over the images are identified and for each image patch the offsets in range and azimuth are determined based on the maximum of the cross-correlation function between the image patches in the two SLCs. However there can be cases in which the accuracy achieved by this co-registration approach is not sufficient. Here we present an alternative co-registration procedure based on a co-registration lookup table between SLCs and a DEM put into the radar geometry of the reference SLC.

The offset model used by the SLC resampling program in the ISP, ***SLC_interp***, is a simple polynomial that is sufficient for most cases where offsets due to topography are less than 0.2 of a pixel. However for higher resolution systems combined with large baselines, such as RADARSAT-1 or even PALSAR, these topography related offsets can exceed 1 pixel. In such case an alternative co-registration approach that is not based on the correlation properties of the two images can be used.

The approach presented in the GAMMA software makes use of a lookup table linking the geometries of the two SLCs based on a DEM of the area. This means that offsets due to topography are incorporated in the lookup table. Another advantage of using a lookup table is that SLC images acquired using different geometries, such ascending and descending tracks, can be co-registered directly to one another.

The procedure is here listed. More details can be found in the following Sub-sections. A practical example is provided in Example I.

1. Generate multi-look images of the reference SLC (here named MLI-1) and the SLC to be resampled (here named MLI-2)
2. Generate a DEM in SLC range-Doppler coordinates based on the geometry of MLI
3. Generate the co-registration lookup table
4. Refinement of initial co-registration lookup table
    1. Transform MLI-1 with the initial lookup table to the geometry of MLI-2
    2. Determine offsets between the resampled MLI-1 and MLI-2 to refine the values of the co-registration lookup table. The procedure is similar to the one described for refining a geocoding lookup table.
5. Calculate the initial resampling of SLC-2 into SLC-1 reference geometry
6. Further refinement of the co-registration by estimation of offsets between the resampled SLC-2 image and the reference SLC. Offsets are estimated using the cross-correlation algorithm described in the User's Guide of the ISP module.
7. New resampling of SLC-2 using the lookup table and offset information. Confirm lookup table offset correction and goodness of fit by simultaneous display of the newly resampled SLC-2 image and the reference SLC as well as offset computation between these two images using the cross-correlation algorithm.

11.1. Generation of multi-look images

The co-registration procedure based on the lookup table requires MLI images as input. Multi-look images of the two images forming an interferometric pair are obtained with the ISP program ***multi_look***.

11.2. Generation of a DEM in radar geometry

The co-registration procedure requires a DEM in radar geometry (see below), which perfectly matches with the reference SAR image. To obtain the height map in radar geometry the procedure including generation of a DEM/map parameter file, generation of a geocoding lookup table, refining of the lookup table and forward geocoding, must be used.

11.3. Generation of the co-registration lookup table

The program ***rdc_trans*** calculates the lookup table to resample MLI-2 into the geometry of MLI-1 where MLI-1 was used to generate a geocoded image of SLC-1. The lookup table has the dimension of the MLI-2 image and contains the range and azimuth pixel numbers of the associated pixel in MLI-1. The program takes into account terrain heights. Since the lookup table is based only on the orbital information contained in the ISP SLC parameter files, the transformation represented in the lookup table is not fully correct. Refinement of the lookup table is therefore necessary.

The look-up table can be displayed with the DISP program ***dismph***. Holes in the lookup table can be filled with the ISP program ***interp_ad***.

11.4. Refinement of the initial co-registration lookup table

The refinement of the co-registration lookup table is done using the same procedure as for the refinement of the geocoding lookup table.

Similarly to the case of the refinement of the geocoding lookup table when at first the simulated SAR image in map geometry was transformed to the geometry of the SAR image in radar geometry, the first step in refining the co-registration lookup table consists in transforming the MLI-2 image to the geometry of the MLI-1 image using the initial lookup table. To transform the MLI-2 to the geometry of MLI-1 use the program ***geocode***. The output consists of the resampled MLI-2 to the MLI-1 geometry. The overlap might however be only partial due to errors in the orbital data used to generate the co-registration lookup table.

To improve the co-registration lookup table the same procedure used for the geocoding lookup table is followed. Instead of the simulated SAR image in radar geometry and the reference SAR image, the procedure is here applied to the resampled MLI-2 image to the master geometry and the reference MLI-1 image. The procedure is summarized below

- Generation of a DIFF/GEO parameter file with the program ***create_diff_par***. The file is here initialized. It will be updated throughout the next processing steps and will contain the offset information between the resampled MLI-2 image and the reference MLI-1 image.
- Computation of initial offsets between the resampled MLI-2 image and the reference MLI-1 image with the program ***init_offsetm***.
- Computation of offsets in small image chips with the program ***offset_pwrm***. Size and number of the image patches depends on number and distribution of features generating contrast in the two images.
- Generation of offset model polynomial with the program ***offset_fitm***. It is suggested to use 3 or 4 coefficients. The offset polynomial is stored in the DIFF/GEO parameter file.
- Refinement of the lookup table using the offset model polynomial with the program ***gc_map_fine***.

11.5. Initial resampling of SLC-2 to reference SLC geometry

With the newly computed co-registration lookup table it is now possible to resample SLC-2 into the geometry of the reference SLC-1. This is done with the program ***SLC_interp_lt***. The input consists of the SLC-2, the ISP SLC parameter files of both SLCs, the ISP MLI parameter files of both MLIs. A slight offset between the resampled and reference SLCs usually remains after this initial resampling due to a combination of inaccuracies in DEM, orbit state vectors, and geocoding, thus requiring a further refinement step of the co-registration lookup table.

The program also supports adding an ISP offset parameter file (OFF_par) that contains range and azimuth offset polynomials. These are not available at this stage and therefore this file is here not considered. An ISP offset parameter will be determined during the second refining stage of the co-registration lookup table between SLC-1 and the resampled version of SLC-2 that we just obtained.

11.6. Further refinement of the co-registration

To improve the co-registration lookup table we can now use the traditional co-registration method based on the cross-correlation of intensities. The resampled SLC-2 image and the reference SLC-1 image are now almost perfectly overlapping and therefore large mis-estimation of offsets is not likely to occur. The procedure to estimate the offsets is summarized below.

- Generation of an ISP offset parameter file with the program ***create_offset***. The file is here initialized. It will be updated throughout the next processing steps and will contain the offset information between the resampled SLC-2 image and the reference SLC-1 image.
- Computation of offsets in small image chips with the program ***offset_pwr***. Size and number of the image patches depends on number and distribution of features generating contrast in the two images. For more information see the User's Guide and the Reference Manual of the ISP module.
- Generation of offset model polynomial with the program ***offset_fit***. It is suggested to use 3 or 4 coefficients. The offset polynomial is stored in the ISP offset parameter file.

11.7. New resampling of SLC-2 to reference SLC geometry

To obtain a new resampled SLC-2 image we use again ***SLC_interp_lt*** this time with offset information contained in the ISP offset parameter file. To check the quality of the new resampled SLC-2 image, the offset computation can be repeated. Looking at the standard deviation information for the fit of the offset model will tell whether if a further resampling is required or if the resampled SLC can be accepted.

The final resampled SLC-2 image and the reference SLC-1 image can then be combined interferometrically using for example the program ***SLC_intf*** as showed in the User's Guide of the ISP module.

12. Co-registration of two MLIs using a lookup table

The procedure is similar to the one described for the co-registration of two SLCs. Generation of the initial lookup table and first resampling are as described in Sections 6 and 8. Then MLI-2 is resampled to MLI-1 using the program ***MLI_interp_lt*** in a similar manner to how SLC-2 was resampled to SLC-1 with the program ***SLC_interp_lt***. To further refine the co-registration the sequence described in Section 11 can be used. Since we are working with MLI images the programs ***create_diff_par***, ***offset_pwrm*** and ***offset_fitm*** have to be used instead of ***create_offset***, ***offset_pwr*** and ***offset_fit***. Finally the MLI-2 image can resampled to the reference MLI image with the program MLI***_interp_lt*** using the offset information stored in the DIFF/GEO parameter file. To check the quality of the new resampled MLI-2 image, this offset computation can be repeated. Looking at the standard deviation information for the fit of the offset model will tell whether if further resampling is required.

The entire procedure can be applied also to PRI images, i.e. images in ground range. First the images must be transformed to slant range geometry. This can be done with the program ***GRD_to_SR*** (part of the ISP module). The co-registered images can be transformed back to the original ground-range geometry with the program ***SR_to_GRD***.

The look-up table can be displayed with the DISP program ***dismph***. Holes in the lookup table can be filled with the ISP program ***interp_ad***.

13. Multi-source image registration

Co-registration of two data sets facilitates the combined interpretation of the data. As a part of the software, tools to co-register data sets of similar but not equal geometry to identical geometry are included. This includes SAR data of different frequencies, incidence angles and orbits or geocoded satellite SAR and optical data of similar resolution (e.g. Landsat TM, ASTER). Automated fine registration based on real valued or complex valued data and manual fine registration based on tie points selected by an operator are supported.

In the geocoding sequence, this registration step is required for the fine registration of an image which was geocoded based on the initial geocoding lookup table but does contain slight offsets due to imperfect orbit data. Typically for the refinement a simulated SAR image from a DEM transformed to the radar geometry and the actual SAR image to be geocoded are co-registered. However, it can happen that it is not possible to find enough samples for obtaining an accurate co-registration polynomial in which case it could be better to resample the geocoded SAR image to the one of a well-geocoded image (optical or radar).

In differential interferometry this registration step is required for:

- The registration of a simulated multi-look SAR intensity image to a real multi-look SAR intensity image (2-pass differential interferometry)
- The registration of two unwrapped phase images (4-pass differential interferometry)

Below the steps for multi-source image registration are described in more detail. For an example see Example E. The assumption here is that we have two geocoded images in different map projections.

13.1. Transformation of one image to the geometry of the reference image

At first the image to be resampled to the reference image needs to be transformed to the same projection and pixel spacing of the reference image. This is achieved with the programs ***dem_trans*** (for a DEM) or ***map_trans*** (for any other image). When running these the user can generate a lookup table (initial) that contains the transformation between the two geometries. This should not be confused with the geocoding lookup table generated when geocoding the SAR image.

13.2. Co-registration of the two datasets

Co-registration between the two images is required if the two images are not yet perfectly overlapping. The co-registration of real or complex valued data to identical geometry consists of two main steps. In the first step the registration function is determined. In the second step the registration function is applied in the resampling of the data to the common geometry.

In the automated fine registration approach the registration function is determined in an initial offset estimation followed by a cross correlation analysis for a large number of image segments, and the determination of bilinear registration offset polynomials.

The manual fine registration approach is more flexible in the way that an "analog" reference data set such as a topographic map may be used. In a first step an operator selects tie points and determines registration offsets for the tie points. These values are then used to calculate the bilinear registration offset polynomials.

The automated co-registration procedure is based on the cross-correlation of intensities. The program sequence consists of generation of a DIFF/GEO parameter file with ***create_diff_par***, computation of offsets in small image chips with ***offset_pwrm*** and generation of the offset polynomial with ***offset_fitm***. The sequence ***offset_pwrm / offset_fitm*** can be repeated a couple of time to improve the quality of the estimate, because the bilinear co-registration function is used to guide the search of the positions where cross-correlation is applied.

If this method is not satisfactory, a list of control points can be manually selected with ***gcp_2ras*** and used to determine the fine registration polynomial with ***offset_list_fitm***.

13.3. Resampling

Once the registration function is known both real and complex valued data sets can be resampled to the reference geometry.

If the user created a lookup table, the bilinear fine registration polynomial is first used to refine the lookup table. For this use the program ***interp_cpx***. For the interpolation nearest-neighbor, sinc, and biquadratic spline interpolation algorithms are included. Resampling of one image to the reference is then performed with the program ***geocode_back***. The refined lookup table is used to transform the original dataset to resample to the geometry of the reference image.

The refinement of the co-registration between the two images can also be applied the directly without the need of a lookup table. In this case the resampling is applied to the image obtained with ***dem_trans / map_trans*** and not to the original image as in the previous case. For the resampling of real valued data the program ***interp_real*** is available. The dataset obtained with ***dem_trans / map_trans*** is resampled to the reference geometry by 2-D interpolation. The user may select between a sinc interpolation algorithm and a nearest neighbor algorithm. The nearest neighbor algorithm is preferred for the resampling of real valued data sets which include gaps, such as the unwrapped phase image which usually includes areas which are set to the NULL value because no unwrapped phase was obtained in the phase unwrapping. For most other cases the sinc interpolator resampling algorithm is preferred. For the resampling of complex valued data the program ***interp_cpx*** is available. The dataset obtained with ***dem_trans / map_trans*** is resampled to the reference geometry using a 2-D complex interpolation. The user may select between nearest-neighbor, sinc, and biquadratic spline interpolation algorithms.

14. Measuring range and azimuth offsets on a regular grid in radar geometry

In the User's Guide of the ISP module the offset tracking method was described. Offsets between SLCs were measured, which were then converted to displacements in the range and azimuth direction. The method was described when two SLCs were available. The method can be applied also if two MLI images are available. For a general description of the method please refer to Example C in the User's Guide of the ISP module. Below a short summary is provided with explanation of the commands used if doing offset tracking with MLI images.

First the offsets between the two MLI images are computed using several image chips uniformly distributed over the images. For this the sequence ***create_diff_par***, ***init_offsetm***, ***offset_pwrm*** and ***offset_fitm*** shall be used. The resulting registration offset polynomials for range and azimuth offset are written to the DIFF/GEO parameter file created with ***create_diff_par***. The user can judge the quality of the registration between the images by looking at the estimated standard deviation of the offsets in range and azimuth.

Once the bilinear polynomial function is known, many offsets are estimated in the area of interest, again based on the image intensity cross-correlation. For this use the program ***offset_pwr_trackingm***. The bilinear polynomial function obtained at the previous stage serves as indication for the position where to estimate the precise offsets. The field of registration offsets here obtained is stored in a new FCOMPLEX file *.offs, where the real part corresponds to the offsets in range and the imaginary part to the offsets in azimuth. The corresponding quality measure field is stored in a new file *.snr. The area where the offsets are to be estimated and the number of estimates can be specified on the command line. For more details see the DIFF/GEO Reference Manual and the description provided for the similar ***offset_pwr_tracking*** program in Example C in the User's Guide of the ISP module.

The bilinear polynomial function determined all over the image and stored in the *.off file is used for the separation of the orbital offsets from those of the area concerned with displacement. The range and azimuth offsets (file *.offs) generated by ***offset_pwr_trackingm*** are transformed in range and azimuth displacements by means of the program ***offset_trackingm***.

The output displacements are written to a binary file in FCOMPLEX format and (optionally) to a text file. The displacements can be computed in range and azimuth directions in pixels or meters or in ground-range and azimuth directions (horizontal geometry) in meters. Points can be rejected based on a SNR thresholding. Offset estimates with a SNR value below the indicated threshold are not considered.

The output file in FCOMPLEX format can be transformed in files of float with the program ***cpx_to_real*** (LAT module) in order to obtain the real or imaginary parts only (i.e. the range and azimuth displacements) or the maximal intensity or phase (i.e. direction) of the displacement. These files can be represented with the DISP programs ***dishgt*** or can be saved to a SUNraster / bmp image file with ***rashgt***. The program ***multi_real*** can be used to generate a multilook intensity image that has the same geometry of the offset field.

The offset parameter file as output to ***multi_real*** can be also used for geocoding of the results with ***gc_map***. Instructions for geocoding of SAR images can be found in Sections 5 to 9.

15. Measuring range and azimuth offsets on a regular grid in map geometry

It is possible using the DIFF/GEO software to measure range and azimuth offsets on a regular grid in a map geometry. These offsets can be used for displacement mapping for example of glaciers. Below the processing sequence is described. An example is given in Example G.

To start with a DEM parameter file describing the map projection used must be defined. For this use the program ***create_dem_par***. To identify the area automatically the user can specify the ISP SLC parameter file of the SAR image for which the offsets shall be measured. Given the DEM parameter file a lookup table for the geocoding of the SAR image needs to be generated. For this use one of the programs listed in Section 6 depending whether a DEM is available or not and whether the SAR image is in slant or ground range geometry (***gc_map, gec_map, gc_map_grd, gec_map_grd***). If refinement of the lookup table is possible, it should be considered. The SAR image must then be geocoded using ***geocode_back*** and a SUNraster version of it be produced with the DISP program ***raspwr***.

Offsets should only be measured at points in the map geometry with image coverage. Furthermore, the user may want to restrict offset estimation to a smaller patch in the geocoded image. The program that generates the list of coordinate pairs where to measure offsets is ***dem_RDC_list***. This program generates coordinate lists in Range-Doppler Coordinates (RDC) and DEM coordinates using a lookup table. The program can use a raster image that serves as a mask. This mask must be an 8-bit SUN raster or BMP image. If the value of the color table entry associated with the pixel is other than black (RGB=0,0,0), it is included in the regions where offsets can be measured.

The coordinate lists are saved in two text files here named as clist_RDC and clist_MAP consisting of pairs of numbers denoting the column and row coordinates in the radar and the map geometry respectively. The next step is to generate an image of the points in the geocoded MLI raster image using ***ras_clist*** to plot the patch center coordinates. An example is provided in Figure 13. The same points can be plotted in the RDC image using the clist_RDC values and the SUNraster file created from the MLI image in radar geometry.

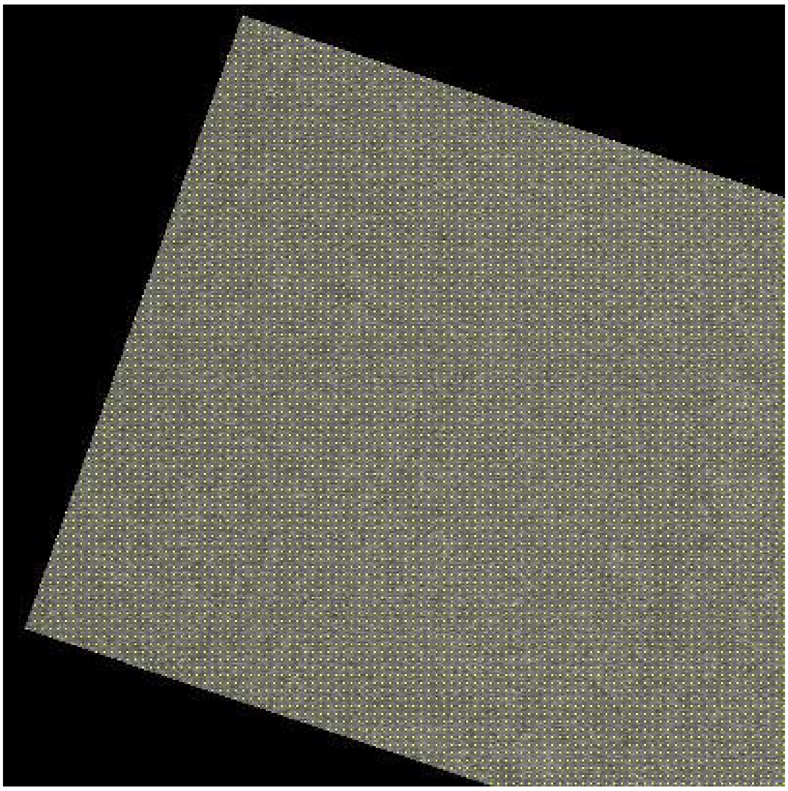


Figure 13. Section of a RADARSAT-1 geocoded image in SUNraster format showing offset measurement locations in map coordinates.

Measuring the offsets requires an offset parameter file to contain any linear a priori offset model. These model coefficients can be estimated in RDC coordinates using the results from ISP program ***offset_pwr***. Offsets are then measured at the RDC coordinates stored in the clist_RDC file with the program ***offset_pwr_list***. This program works similarly to the ***offset_pwr*** program. The output is written to an offset_clist file in text format and to the offs_clist and snr_clist files as binary floating point numbers.

Displaying the offset information can be done by conversion of the FCOMPLEX format output raster files to separate range and azimuth offset files and displayed using the DISP program ***rashgt*** with a wrap of some pixels (e.g. 3).

16. Additional tools for geocoding and image registration

16.1. Generation of parameter files for already geocoded images

Recently for a number of sensors image products in geocoded format are directly available. The GAMMA Software supports geocoded ALOS PALSAR, TerraSAR-X and COSMO-SkyMed products.

For Level 1.5 products produced by EORC/JAXA in CEOS format, the program ***par_EORC_PALSAR_geo*** generates the ISP image parameter file from the CEOS metadata and a DEM/MAP parameter file if the image has been provided in geocoded format. The program also reformats the image data. In this way the image data is compatible with the GAMMA software. Data are already fully calibrated.

For TerraSAR-X EEC data produced by DLR, the program ***par_TX_geo*** reads the TerraSAR-X data and the annotation file (*.xml in the main directory of the image data) as provided by DLR and creates the ISP SLC image parameter file, the DEM/MAP image parameter file and the image data file in the format used by GAMMA software. For multi-polarization data the specific program has to be repeated for each single channel. The annotation file is the same for all channels. The output consists of fully calibrated data.

For COSMO-SkyMed GTC data produced by ASI, the program ***par_CS_geo*** reads the image data file (in HDF5 format) and creates 1) the ISP SLC image parameter file, 2) the DEM/MAP image parameter file and 3) the image data file in the format used by GAMMA software. The data are already relatively calibrated. The program transforms the amplitude data to intensity and applies the calibration constant provided in the header information.

Processing examples

In this part a list of processing examples dealing with different aspects of image geocoding are presented.

It should be remarked that parameter values provided in the processing example cannot be considered valid for all cases. It is possible that one or more values might have to be adapted to the specific case being processed. It is advised to look carefully at the messages printed on stdout when running each individual program. For assistance please get in contact with us (gamma@gamma-rs.ch).

Example A - Preparation of DEM for geocoding with DIFF&GEO
Steps for the preparation of a DEM to be used in the GAMMA Software for geocoding are presented.

Example B - GTC-Geocoding with DEM in map coordinates
Processing sequence for terrain geocoding using a DEM and a simulate SAR intensity image

Example C - GEC-Geocoding without DEM
Processing sequence for geocoding without a DEM

Example D - GTC geocoding with InSAR heights
Processing sequence for geocoding using an InSAR-based DEM

Example E - Multi-source image registration with Landsat image
Processing sequence for co-registration of geocoded images in different projection, map geometries etc. – Landsat image in binary format

Example F - Multi-source image registration with ASTER image
Processing sequence for co-registration of geocoded images in different projection, map geometries etc. – ASTER image in SUNraster

Example G - Geocoding using an external reference image
Processing sequence for terrain geocoding of a SAR image using an external reference image

Example H - Measuring offsets on a grid
Processing sequence for offset estimation on a grid of points

A. Preparation of DEM for geocoding with DIFF&GEO

This Annex guides the user through the steps required for the preparation of a DEM to be further used for geocoding a SAR image. Preparation of the DEM includes the generation of a DEM parameter file, improvement of the DEM (e.g. correction for missing values) and transformation between map projections.

To guide the user through these processing steps a tile of SRTM-3 DEM data will be used (see Section A.1 for details on this type of DEM). A note on mosaicing SRTM DEMs is given at the end of this Annex

A.1. Introduction

Digital Elevation Models can be available from many sources (satellite data, ground survey, topographic maps, SAR interferometry etc.). If a DEM is not available, two (nearly) global datasets that can be considered suitable for geocoding currently available spaceborne SAR data are the freely available SRTM and GTOPO30 DEMs.

Freely available SRTM (Shuttle Radar Topography Mission) DEMs are at 3 arcsec resolution (~ 90 m) for all land masses between 60 degrees N and S (SRTM-3) and at 1 arcsec resolution (~ 30 m) for the US. These have been obtained from the C-band data acquired by the SRTM instrument.

The basic SRTM data (Version 2) is available at http://www2.jpl.nasa.gov/srtm/ in tiles each covering 1 degree in longitude and 1 degree in latitude. Each tile is 1201 pixels wide and 1201 pixels long, and has posting of 8.3333330e-04 degrees. Each tile is identified by the geographical coordinate of the bottom left pixel. For example a tile called N10E20.hgt covers an area between 10 and 11 deg N and between 20 and 21 deg E. An improved version of the SRTM-3 DEM is available under http://srtm.csi.cgiar.org/. In this case tiles are larger compared to the Version 2 tiles. All datasets are in equiangular projection, short integer, big-endian byte format (i.e. most important byte first).

Additionally (but not freely available) X-band DEM with 1 arcsec posting can be obtained from the German Aerospace Agency, DLR.

GTOPO30 is a global DEM available from the US Geological Survey (USGS) with 30 arcsec resolution (~ 1 km). This is the only DEM currently available on a continental scale for areas above 60 N and below 60 deg S. The low resolution hinders a very accurate geocoding of high resolution data. However it can be considered suitable when geocoding low resolution SAR data such as ASAR Global Monitoring mode. GTOPO30 DEMs are in equiangular projection, short integer, big-endian byte format (i.e. most important byte first). Data is available at http://edc.usgs.gov/products/elevation/gtopo30/gtopo30.html.

A merged global DEM from SRTM-3 and GTOPO30 is also available at http://www2.jpl.nasa.gov/srtm/

A.2. Generation of DEM parameter file

The DEM used in this example is a Version 2 SRTM-3 DEM tile called N36W116.hgt. It covers an area between 36 and 37 degrees North and 116 and 115 degrees West, which corresponds to the area of Las Vegas, Nevada. The tile is 1201 pixels wide and 1201 pixels long, is in short integer format (2 byte per pixel) and has the same posting of 8.3333330e-04 degrees in latitude and longitude. The DEM is available on the DEMO-CD in the dem directory. Copy it to your working directory to run this example.

In order to be able to use a DEM in the GAMMA software a DEM parameter file has to be generated. The generation of the DEM parameter file is supported by the program ***create_dem_par***, which requires the user to input a series of parameter values. Alternatively, the user can save to a text file all the values to be given as input on the command line, from which ***create_dem_par*** will then read the values automatically.

```
create_dem_par N36W116.dem_par
```

The user is asked to provide some parameter values on the command line

Definition of DEM projection
Several projections are available. In this example the DEM is in equiangular format, which means that the value EQA has to be given in input. However, since EQA is also default value, the user can simply press the enter key.

Definition of DEM Datum, Ellipsoid, and Map Projection
The user is requested to provide the name of a country at which a list of available datums and ellipsoids will be printed on screen. If the WGS-84 datum shall be used, the string WGS84 can be used directly. In this case we use WGS72.

Definition of additional parameters

- DEM title: a string that identifies the DEM parameter file (default is DEM). Here we use SRTM_dem.
- data format: the format of the image (float or short integer, default is float, i.e. REAL*4). Here we use INTEGER*2.
- DEM height offset (m): to be defined if the DEM has an offset with respect to the 0 m level (default is 0 m). Here we use 0.0.
- DEM height scale factor: to be defined if the DEM is scaled (default value is 1.0). Here we use 1.0.
- DEM width: number of samples of the DEM on one line. Here we use 1201.
- DEM length: number of lines of the DEM. Here we use 1201.
- Posting: pixel size. If EQA projection chosen the posting will be given in decimal degrees otherwise in meters. Here we use -8.333333e-04 for the latitude posting and 8.333333e-04 for the longitude posting. The latitude/northing posting is always negative since the latitude/northing decrease for increasing line number (i.e. when going from the top of the bottom to the DEM). The longitude/easting posting is always positive since the longitude/easting increase for increasing column number (i.e. when going from the left to the right part of the DEM).
- Northing/easting or latitude/longitude of first DEM sample: coordinates of the top left corner of the DEM. Here we use 37.0 for the latitude and -116 for the longitude.

The DEM parameter file called N36W116.dem_par can be opened with a text editor or displayed on the screen. Errors in the DEM parameter file can be corrected in a text editor.

```
Gamma DIFF&GEO DEM/MAP parameter file
title:  N36W116.hgt
DEM_projection:    EQA
data_format:       INTEGER*2
DEM_hgt_offset:         0.00000
DEM_scale:              1.00000
width:            1201
nlines:           1201
corner_lat:    37.0000000  decimal degrees
corner_lon:   -116.0000000  decimal degrees
post_lat:   -8.3333330e-04  decimal degrees
post_lon:    8.3333330e-04  decimal degrees

ellipsoid_name: WGS 72
ellipsoid_ra:       6378135.000   m
ellipsoid_reciprocal_flattening:  298.2600000

datum_name: WGS 1972
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, WGS72, World
```

To display the DEM we can use the program ***disdem_par*** of the DISP module. This program displays the DEM as shaded relief. It uses the DEM parameter file to associate to each pixel the corresponding coordinates in the chosen map projection as well as the values of latitude and longitude.

```
disdem_par N36W116.hgt N36W116.dem_par
```

In this way the DEM can also be checked for missing values and gaps. As shown in Figure A1, which represents a screenshot of the DEM display, the DEM presents some gaps. In SRTM DEM missing values appear as -32768 values.

A.3. Correction for missing values

Gaps of data can be eventually filled by interpolating over surrounding areas. With the GAMMA software this is done in a number of steps, which are described below.

While in a DEM typically the value 0 represents elevation of 0 m, in the GAMMA software the value 0 is used for missing values. For this reason, all pixels with 0-m elevation have to be treated in order to avoid that they are flagged as gaps in the DEM. A simple way to deal with this issue is to replace all zero values with a similar value, e.g. 1 in the case of short integer data or 0.1 in the case of real-valued data. For this we use the DISP program ***replace_values*** and generate a dummy DEM called temp_dem in which all 0 values have been replaced with 1.

```
replace_values N36W116.hgt 0 1 temp_dem 1201 0 4
```

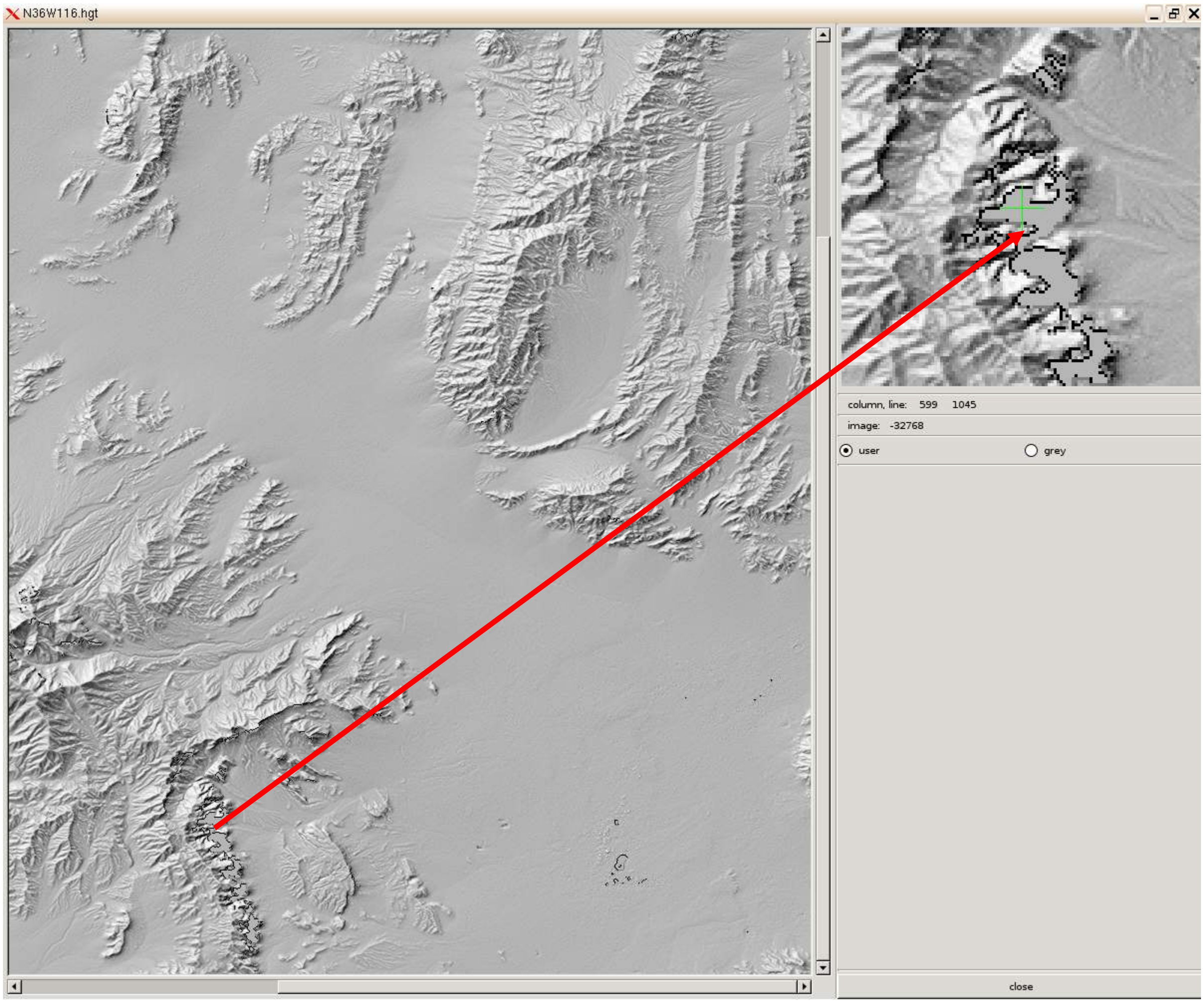


Figure A1. Screenshot of DEM represented as shaded relief using the program disdem_par. The display highlights an area with missing values (equal to -32768).

Now all missing values in the (dummy) DEM can be replaced by missing values in the GAMMA software. In other words we can now replace all values equal to -32768 (no data) with 0. For this the program ***replace_values*** is used and a new dummy DEM file called temp_dem2 is generated.

```
replace_values temp_dem -32768 0 temp_dem2 1201 0 4
```

Finally we interpolate over areas with zero values using either the program ***interp_real*** or, better, the program ***interp_ad*** (which is part of ISP module). It should be noted that this procedure is fine for small holes. The user can choose the size of the interpolation window and the min/max number of samples to be used for interpolation. In our example the command line looks as follows. In the end we generate a new DEM called N36W116.dem.

```
interp_ad temp_dem2 N36W116.dem 1201 9 40 81 2 4
```

The new DEM can be displayed as shaded relief with ***disdem_par*** as follows

```
disdem_par N36W116.dem N36W116.dem_par
```

A.4. Mosaicing SRTM DEMs

If the area to be geocoded covers an area that is included in more than one SRTM tile, several tiles can be mosaiced together to obtain a larger DEM.

The steps to be performed to obtain a mosaiced DEM consist of

- Generation of DEM parameter file for each SRTM tile with the program ***create_dem_par*** (see Section A.2)
- Mosaicing with the programs ***mosaic*** or ***multi_mosaic*** (part of the LAT module – see the User's Guide of the LAT module for more information)
- If needed, correction for missing values with the procedure described in Section A.3.

Depending on the size of the area to be geocoded, one can use either this procedure or directly one tile of the Version 3 SRTM DEM described in Section A.1.

A.5. Transformation of map projection

Typically geocoding of an image is preferred to a projection that preserves dimensions. For this reason when working with SRTM DEM it is necessary to transform the DEM from the equiangular projection to another map projection. To transform a DEM from a given projection to another projection the following steps must be performed

- Generation of a DEM parameter file for the new projection. This is done with the program ***create_dem_par***.
- Transformation of the DEM from the given projection to the new projection. This is done with the program ***dem_trans*** that reads in the original DEM parameter file, the new DEM parameter files and the available DEM, and establishes the transformation between the two projections in form of a lookup table. The output is the DEM in the new projection.

A practical example is given in Example B.

B. GTC-Geocoding with DEM in map coordinates

For terrain corrected geocoding a digital elevation model has to be available. If a DEM in map coordinates is available terrain geocoding of an image in slant range geometry requires the following steps:

- DEM/MAP parameter file creation using the program ***create_dem_par*** and/or eventually transformation of the DEM to another projection. This is done with the program ***dem_trans***.
- Derivation of initial geocoding lookup table that links map geometry to range-Doppler geometry, and simulation of a SAR intensity image in map geometry based on the DEM and the SAR imaging geometry. This is done using the program ***gc_map***.
- Resampling of the simulated SAR intensity image from map to range-Doppler radar geometry using the program ***geocode*** to conduct a forward transformation based on the initial geocoding lookup table.
- Fine registration of the simulated intensity image with the real SAR image intensity (***create_diff_par*** followed by ***init_offsetm***, ***offset_pwrm***, and ***offset_fitm***). The offset registration polynomials in the DIFF parameter file are then used to refine the geocoding lookup table (***gc_map_fine***). As an alternative approach, a list of control points may be manually selected using gcp_ras from the SAR image and a map and used to determine the offset registration polynomials in the DIFF parameter file (***create_diff_par*** followed by ***offset_list_fitm***).
- The resulting refined lookup table is used for the (backward) geocoding from SAR to map coordinates (***geocode_back***) and (forward) geocoding from map to radar geometry (***geocode***).

Terrain corrected geocoding of SAR images in ground-range / azimuth coordinates follows the same processing sequence, the only difference being in the program that generates the initial geocoding lookup table. This is supported by the program ***gc_map_grd***.

The Table below summarizes these steps highlighting the programs required

Step	**Program(s) used**
1. DEM/MAP parameter file creation	***create_dem_par***
2. Derivation of initial geocoding lookup table and SAR intensity image simulation	***gc_map (gc_map_grd)***
3. Transformation of simulated SAR intensity image from map to radar geometry	***geocode***
4. Fine registration	***create_diff_par, init_offsetm, offset_pwrm, offset_fitm***
5. Refinement of initial geocoding lookup table	***gc_map_fine***
6. Backward geocoding from SAR to map coordinates and forward geocoding from map to radar geometry	***geocode_back, geocode***

B.1. Introduction

To demonstrate GTC geocoding with a DEM we consider an ERS-1 image acquired over Las Vegas on 23 May 1996. As file identifier we use the orbit number, 25394. The image is in SLC format. Aim is geocoding the SAR intensity image to UTM projection with 25 m posting. As DEM we use a mosaic of SRTM-3 tiles with in equiangular projection with posting ~ 90 m.

The following files are used in the processing example. All files are available on the DEMO CD-ROM. The SAR image files are in the slc directory. The DEM files are in the dem directory.

Filename	Content
25394.slc	SLC image
25394.slc.par	SLC parameter file
Nevada_srtm_eqa.dem	DEM (mosaic of SRTM tiles) in EQA projection
Nevada_srtm_eqa.dem_par	DEM parameter file

Geocoding will consist of the following processing steps

- Preparation of the MLI intensity image (e.g. if SLC image available or if resolution different than geocoding posting). Generation of ISP/SLC parameter file and orbits improvement, if possible, must be carried out before generating the MLI image.
- Transformation of DEM from equiangular to UTM map projection
- Derivation of initial geocoding lookup table and simulated SAR intensity image in map geometry
- Resampling of simulated SAR intensity image from map to radar geometry (forward geocoding)
- Fine registration of simulated SAR intensity image with the real SAR intensity image
- Refinement of geocoding lookup table
- Geocoding (backward) of SAR intensity image to map geometry

The processing is also supported by an automated processing script in the DEMO-CD scripts directory (run_GEO_LasVegas). The list of commands in this script can be found in the file com_GEO_LasVegas.

The script should be considered as an introduction to scripting and can be used for developing own scripts based on the user's particular needs. If the script is used for processing, it is strongly recommended to adapt it by selecting the programs actually needed for processing and by critically choosing the values of the parameters required by each individual program. For this purpose it is highly recommended to refer to the Reference Guide.

B.2. Generation of MLI intensity image

When geocoding a SAR image 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. With respect to this issue, there are two situations that can occur.

- If we have a SAR image at higher resolution (e.g. 20 m) and want to geocode it to a lower resolution (e.g. 50 m), direct geocoding would keep the noise unnecessarily high.

In this case it is recommended to first apply multi-looking (or filtering) to reduce the spatial resolution to a similar value as required for the geocoded image and then geocode.

-
- If we have a SAR image at lower resolution (e.g. 100 m) and want to geocode it to a higher resolution (e.g. 25 m), direct geocoding efficiency would be much reduced without much information gain. In this case it is recommended to geocode at a lower resolution and then to oversample the geocoded image to the required higher resolution.

If we have an SLC image available, the MLI intensity image is obtained with the program ***multi_look***. Multi-look factors depend on satellite range/azimuth pixel ratio and on the geocoded product resolution. For example if an ERS or ENVISAT ASAR SLC is available multi-look factors of 1 and 5 in range and azimuth respectively will generate a MLI image with 20-m pixel spacing. In this way the pixel size would be comparable to the posting required for geocoding. The MLI used in this example has been obtained in this way.

```
multi_look 25394.slc 25394.slc.par 25394.mli 25394.mli.par 1 5
```

The MLI intensity image can be displayed with the DISP program ***dispwr***

```
dispwr 25394.mli 2500
```

The width of the SAR intensity image is 2500 pixels. This information is stored in the *.mli.par file obtained when running ***multi_look***. Figure B1 shows the MLI intensity image.

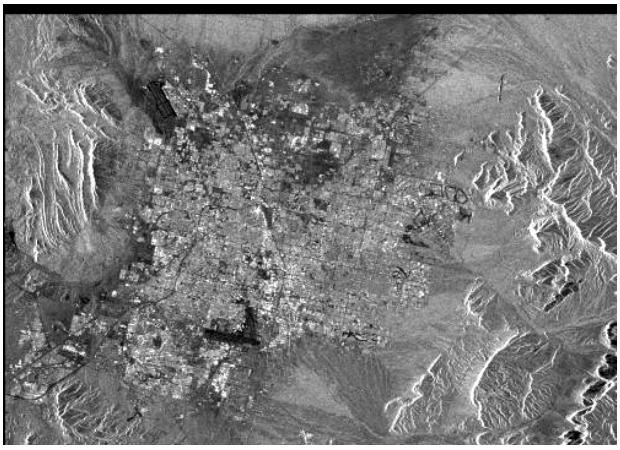


Figure B1. Multi-look intensity image 25394.mli (1800 pixels long and 2500 pixels wide, pixel size 20 m in ground range and azimuth direction).

B.3. Transformation of DEM projection

Transformation of DEM projection has been briefly described in Example A, Section A.5. At first the DEM parameter file for the new projection has to be created. In this way the link between the two projections can be established and the DEM can be resampled from the original to the new projection.

To create the new DEM parameter file the program ***create_dem_par*** must be used. The command line looks as follows

```
create_dem_par Nevada.dem_par 25394.mli.par
```

The DEM parameter file describing the DEM in UTM projection is called Nevada_srtm_utm.dem_par. In this case we also use the parameter file of the intensity image to aid locating the DEM geographically.

Since this is an interactive program. the user is asked to input some information to create the parameter file. Below we summarize the values to be selected. For more information on the creation of a DEM parameter file, refer to Example A, Section A.2.

```
- Select DEM projection -
DEM projection (EQA, UTM, OMCH, …): EQA : UTM

- Select datum-
enter country name: USA
number of available datums for country: USA : 5

ID:    1  DATUM: North American 1927   ELLIPSOID: Clarke 1866
…
ID:    5  DATUM: WGS 1984   ELLIPSOID: WGS 84 REGION: Global Definition, WGS84, World
select DATUM ID number in the range 1 - 5: 5

-Select projection-
input UTM projection parameters : proposed UTM zone number: 11
UTM zone number: (11): 11

- Selection of further parameters
DEM title: DEM : Nevada_srtm_utm
data format(REAL*4, INTEGER*2): REAL*4 : INTEGER*2
DEM height offset (m) (nominal=0.0):       0.000: 0.0
DEM height scale factor (nominal=1.0):    1.00000: 1.0
DEM width (samples):   1526: 1500
DEM length (lines):    1299: 1500
posting (northing (m), easting (m)):   -50.000000    50.000000 : -100.0  100.0
offset of first DEM sample (northing (m), easting (m)):  4036000.000   626000.000 : 4036000.000   626000.000
```

It should be noticed that we have decided to transform the DEM to UTM projection with a 100-m posting, i.e. a size similar to the DEM in the EQA projection.

The size of the DEM in the new projection is suggested by the program and is based on the default posting values (50 m in this case) and the SAR intensity image corner coordinates. Here we decided to use the values 1500 and 1500 because of the larger posting (100 m) and in order to be sure to include the whole area covered by the SAR image within the transformed DEM.

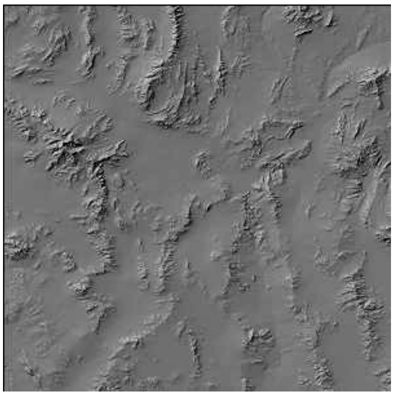
The provision of the SLC parameter file as input allows the programs to compute the corner coordinates of the DEM in the new projection based on the area covered by the SLC.

Errors in the DEM parameter file resulting from incorrect typing can be corrected by opening the DEM parameter file with a text editor. This also allows modifying one or more parameters in case the transformation between projection has not been successful (e.g. part of area covered by SAR image not included as a result of too small width/length or offset of first DEM sample).

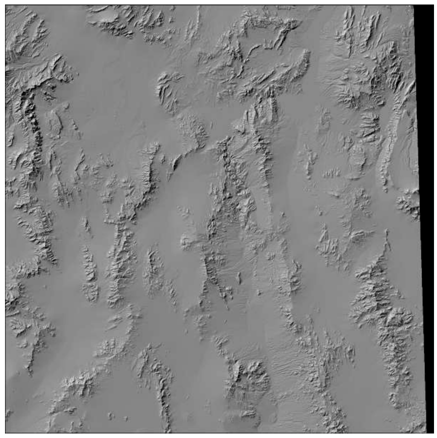
To transform the DEM from EQA to UTM projection we use the program ***dem_trans***. In this example the command line looks as follows

```
dem_trans    Nevada_srtm_eqa.dem_par    Nevada_srtm_eqa.dem    Nevada.dem_par
Nevada.dem 1 1 0
```

Figure B2 illustrates the DEM in EQA projection with pixel size of approximately 90 m and in UTM projection with pixel size of 25 m. Since we defined the corner coordinates of the output DEM on the basis of the SLC parameters, only the part of the DEM that widely includes the SAR image has been transformed.



(a)



(b)

Figure B2. (a) SRTM DEM in EQA projection with ~ 90 m posting and (b) in UTM projection with 100 m posting.

B.4. Generation of initial geocoding lookup table

Once the DEM and the SAR image to be geocoded are ready the initial transformation between the radar and the map geometry can be established, i.e. the geocoding lookup table can be generated. To generate the initial lookup table the program ***gc_map*** must be used. The basic command line in this example looks as follows

```
gc_map   25394.mli.par   -   Nevada.dem_par   Nevada.dem   LasVegas.dem_par
LasVegas.dem LasVegas.rough.utm_to_rdc 4 4 LasVegas.utm.sim_sar
```

If additional geometric product such as the local incidence angle map, the pixel area normalization factor map etc. shall be produced the corresponding file names have to be added.

In this example the program ***gc_map*** produces

- the DEM segment covering the SAR image (LasVegas.dem) and corresponding DEM parameter file (LasVegas.dem_par)
- the initial geocoding lookup table (LasVegas.rough.utm_to_rdc)
- the simulated SAR image in map geometry (LasVegas.utm.sim_sar)

By looking at the DEM parameter file for the DEM segment we can see that the segment is fairly smaller than the initial DEM, being 2376 pixels wide and 1848 pixels long.

Each product can be displayed with a DISP program. More specifically for the DEM we can use ***disdem_par*** or ***dishgt***, for the complex-valued lookup table we can use the program ***dismph*** and for the simulated SAR image we can use the program ***dispwr***. The command lines look as follows

```
disdem_par LasVegas.dem LasVegas.dem_par

dismph LasVegas.rough.utm_to_rdc 2376

dispwr LasVegas.utm.sim_sar 2376
```

The lookup table and the simulated SAR image are displayed in Figures B3 and B4.

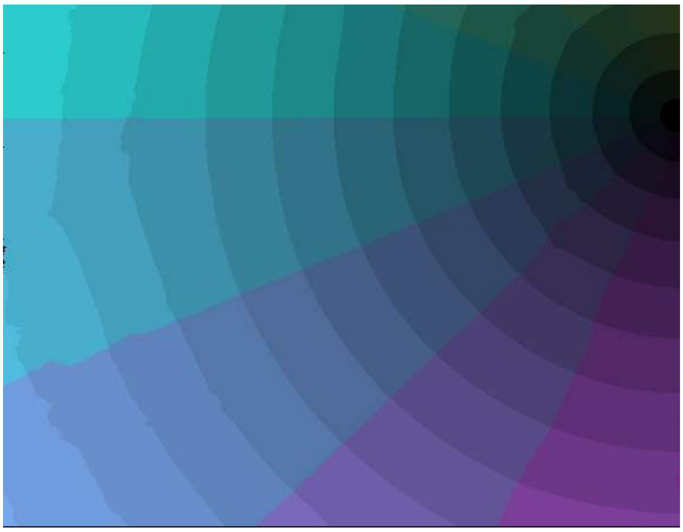


Figure B3. Initial lookup table defining the transformation between map and radar geometry. The lookup table has the same size of the DEM segment and related geometric products.

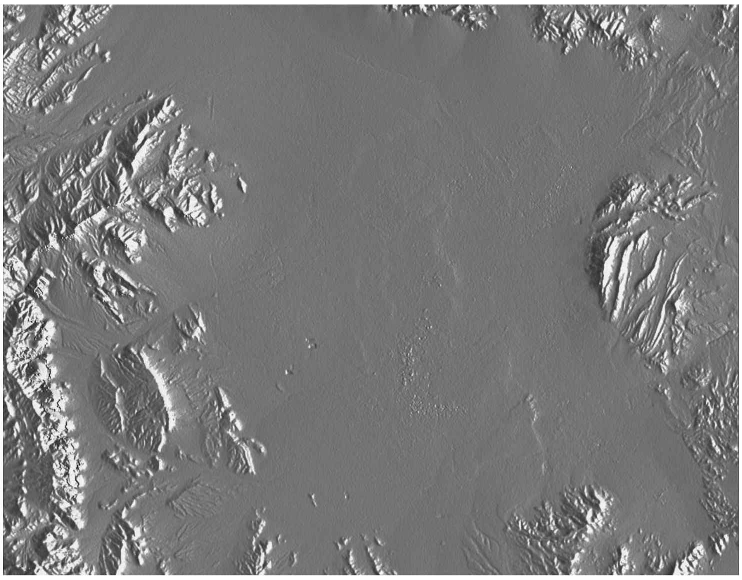


Figure B4. Simulated SAR image from DEM in map geometry (UTM, pixel spacing 25 m).

The lookup table as it is has errors due to imprecision of the state vectors. If this lookup table were used in either the forward or the backward transformation, the result would be affected by offsets with respect to the true geometry. These offsets can however be used to refine the lookup table.

The steps to refine the lookup table are therefore the following
- Transform the simulated SAR image from map to radar 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

The output will be a refined lookup table that will allow the geocoding of the intensity image

B.5. Resampling of simulated SAR image to radar geometry

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

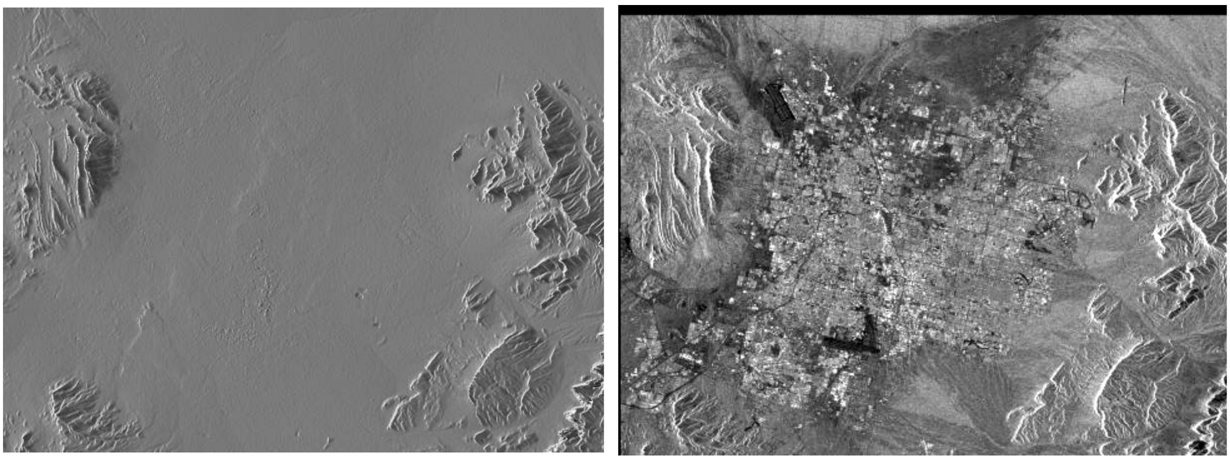
```
geocode LasVegas.rough.utm_to_rdc LasVegas.utm.sim_sar 2376
LasVegas.sim_sar 2500 1800 1 0
```

The output is the simulated SAR image in the radar geometry (LasVegas.sim_sar). To process we need to know the number of columns of the simulated SAR intensity image in the map geometry (2376) and the dimensions of the image to be obtained in the radar geometry (2500 x 1800, i.e. the dimensions of the real SAR intensity image).

To appreciate the offset introduced by the imperfect transformation in the geocoding lookup table the simulated SAR intensity image in the radar geometry can be compared to the real SAR intensity image. This can be done with the DISP program ***dis2pwr:***

```
dis2pwr LasVegas.sim_sar 25394.mli 2500 2500
```

Figure B5 illustrates the two images. Topographic are clearly visible. From a visual inspection the user can estimate the offsets in range and azimuth between the images.



(a) (b)

Figure B5. Simulated SAR intensity image in radar geometry (a) and real SAR intensity image (b).

B.6 Computation of offsets and generation of registration polynomial

The refinement of the geocoding lookup table requires the offsets between the true geometry (real SAR image) and the incorrect geometry (simulated SAR image) to be computed and modeled in form of polynomial.

The first step is the generation of the DIFF/GEO parameter file, which will contain the offsets between the two images and the coefficients of the offset polynomials. To initialize this file we use the program ***create_diff_par***. In this example the command line looks as follows.

```
create_diff_par 25394.mli.par - 25394.diff_par 1
```

This is an interactive program and the user is requested to input a number of values at the command line. Alternatively the values can be read from a text file in which these values have been previously stored. For this it is referred to the script run_GEO_LasVegas on the DEMO CD-ROM.

The file 25394.diff_par can be viewed on screen or with a text editor. Since we have initialized this file, most values are equal to zero.

The computation of the offsets and the generation of the polynomial is similar to the procedure described in the User's Guide of the ISP module for the co-registration of two SLC images.

At first an initial, global, estimate of the offsets in range and azimuth can be computed. This is done with the program ***init_offsetm***. In this example the command line looks as follows.

```
init_offsetm 25394.mli LasVegas.sim_sar 25394.diff_par
```

Here we accepted all default values. It is important that the initial offset estimate does not deviate from the true value. For this reason, if the result is incorrect, it is suggested changing the position of the window used for computing the correlation (for example to an area with topographic features) or its size. If the user does not succeed in obtaining reasonable offsets, it is recommended to set manually the initial offset estimates to 0 and proceed with the computation of the local offsets.

The computation of the local offsets is done with the program ***offset_pwrm***. The local estimates are then used by the program ***offset_fitm*** to generate the co-registration polynomial. The sequence ***offset_pwrm / offset_fitm*** can be repeated a couple of time to improve the quality of the estimate, because the bilinear co-registration function is used to guide the search of the positions where cross-correlation is applied. In a first step, a small number of offsets and an oversampling factor of one can be used with high efficiency. Then, a large number of offsets and oversampling factors of 2 or 4 can be used to improve the quality of offsets and consequently of bi-linear polynomial. The size of the search window shall be adapted to the resolutions of SAR data and DEM. More specifically, number of windows and size of the window depend on occurrence and size of the topographic features in the two images. For SAR/DEM pixel spacing on the order of 25 m, a search window size of 128 or even 256 is appropriate. For higher resolutions (e.g. 100 m) a search window size of 64 is sufficient. Order of the polynomial depends primarily on the length of the scene. A bi-linear function (i.e. 3 coefficients) is sufficient in most cases. For long stripes also the cross-linear coefficient should be considered.

In this example the command lines for the first run of the polynomial generation is as follows

```
offset_pwrm 25394.mli LasVegas.sim_sar 25394.diff_par offs snr 512 512
offsets 1 8 8 7.0

offset_fitm offs snr 25394.diff_par coffs coffsets 7.0 3
```

With ***offset_pwrm*** offsets in range and azimuth between the simulated and the real SAR intensity images are computed in 8 x 8 windows each being 512 x 512 pixels large. Offset values with SNR value above the threshold, which is set to 7.0 are kept. Offsets are saved to the binary file offs, whereas the SNR values are saved to the file snr. The file offsets contains offset information in text format.

If the number of saved offsets is very small it is possible that the number of offsets is not enough to generate a reliable registration polynomial. In this case, one can try to re-run the program with more windows, possibly of different size. The goal is to obtain many windows with similar offsets in order to make the estimation of the offset polynomials more robust.

If the number of topographic features is small, the result will be few offsets satisfying the SNR criterion and therefore a likely incorrect co-registration polynomial. Instead of this automated method a list of control points can be manually selected from the SAR image and the map using the program ***gcp_ras***, which can then be used to determine the fine registration polynomial with ***offset_list_fitm***. If neither this procedure works it is better to abandon the refinement step and to geocode the SAR image directly with the initial lookup table. Positioning errors will be related to the precision of the state vectors.

With ***offset_fitm*** the polynomial with 3 coefficients has been generated. A polynomial with 4 coefficients shall be used in case long and wide stripes of data are being geocoded (e.g. ScanSAR data). The coefficients are written to the *.diff_par file.

To obtain an idea of the quality of the co-registration, we can look at the statistics of the standard deviation of the offsets. Values below 0.3 indicate that geocoding run reasonably well. If both values are higher than it is advised to change one or more of the following parameters: number of windows, size of the window, number of polynomial coefficients, threshold.

The first guess of the offset polynomial can be used in a refinement step, in which ***offset_pwrm*** and ***offset_fitm*** are run again but with slightly different parameter values. For example in this case the command lines look as follows

```
offset_pwrm  25394.mli  LasVegas.sim_sar  25394.diff_par  offs  snr  128  128
offsets 2 24 24 7.0

offset_fitm offs snr 25394.diff_par coffs coffsets 7.0 3
```

Here 24 x 24 windows have been used, each being 128 x 128 pixels large. For the generation of the offset polynomial three coefficients are considered.

In this particular example the standard deviation statistics of the offsets were already good after the first run of ***offset_pwr / offset_fit***. The iteration was shown for demonstrative reason. In fact it did not improve the estimate of the offsets.

The user can verify whether the first coefficient of the offset polynomial in the 25394.diff_par file coincides with the visual guess of the estimate.

B.7. Refinement of geocoding lookup table

To improve the lookup table, the offset polynomials are given as input to the program ***gc_map_fine***. The command line in this example looks as follows

```
gc_map_fine        LasVegas.rough.utm_to_rdc        2376        25394.diff_par
LasVegas.utm_to_rdc 0
```

Once the refined lookup table LasVegas.utm_to_rdc is available the less accurate initial lookup table LasVegas.rough.utm_to_rdc is no longer used.

B.8. Backward geocoding from SAR to map coordinates

Files can now be geocoded from SAR range-Doppler geometry to UTM map geometry using the refined lookup table. The program for backward geocoding is ***geocode_back*** and in this example the command line looks as follows

```
geocode_back 25394.mli 2500 LasVegas.utm_to_rdc 25394.mli.utm 2376 1848 2 0
```

Besides the input and output image file names and the lookup table, the other parameters required are the width of the image in radar geometry (2500) and the width of the image in map coordinates (2376). In this case it is set to be equal to the size of the lookup table. Various interpolation modes are supported. For a SAR intensity image in floating point format the spline-log (option 2) format is the most appropriate one. For SUNraster files only nearest-neighbor can be applied. In this case, attention has to be paid to the left/write flipping of the input data (usually left/right flipping is applied when a SUNraster / bmp file is created in radar geometry).

The geocoded image can be displayed with the DISP program ***dispwr:***

```
dispwr 25394.mli.utm 2376
```

Figure B6 shows the geocoded SAR intensity image.

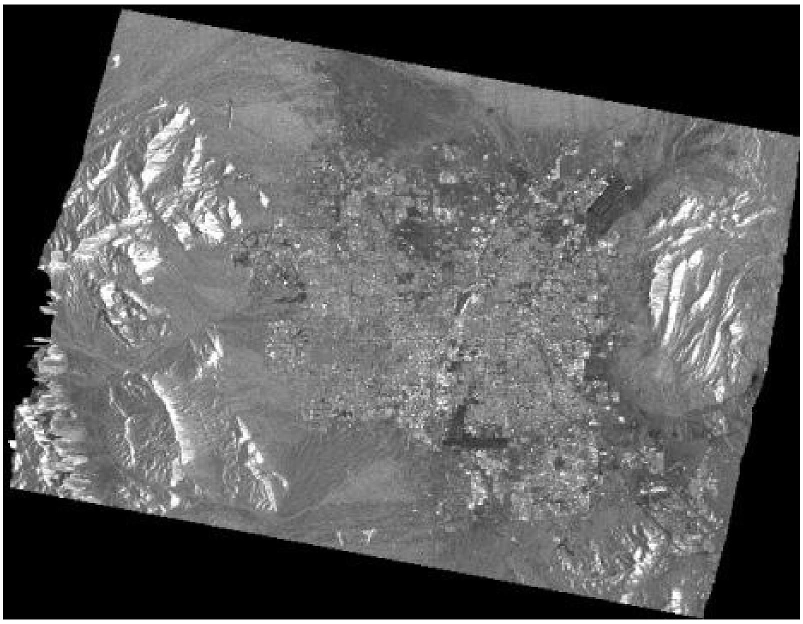


Figure B6. Geocoded SAR intensity image in UTM projection with 25 m posting

It can also be interesting to display the geocoded SAR image and the simulated SAR image in map geometry to appreciate the overlap between the images, i.e. the geocoding accuracy. This can be done with the DISP program ***dis2pwr*** as follows

```
dis2pwr LasVegas.utm.sim_sar 25394.mli.utm 2376 2376
```

B.9. Forward geocoding from map to radar coordinates

The refined lookup table can also be used to transform files (for example the DEM) from UTM map geometry to the radar geometry. This is done with the program ***geocode***. For example the command line for transforming the DEM segment to radar geometry is the following:

```
geocode LasVegas.utm_to_rdc LasVegas.dem 2376 25394.dem 2500 1800 1 0
```

The DEM transformed to radar geometry can be displayed with the dishgt program as follows:

```
dishgt 25394.dem 25394.mli 2500 - - - 160.0
```

One particular reason to transform the DEM to the radar geometry is for the automatic extraction of height control points from the resampled DEM. For more details see the User's Guide of the ISP module. The height control points are of interest for the refinement of the estimation of the interferometric baseline (see ISP ***base_ls*** program).

C. GEC-Geocoding without DEM

No digital elevation model is required for ellipsoid corrected geocoding. Instead of using a variable DEM a constant height value is used for the GEC-geocoding. In this case the processing steps are as follows

- DEM/MAP parameter file creation using the program create_dem_par
- Derivation of the geocoding lookup table that links map geometry to range-Doppler geometry,
- The lookup table is used for the (backward) geocoding from SAR to map coordinates (geocode_back) and (forward) geocoding from map to radar geometry (geocode).

Terrain corrected geocoding of SAR images in ground-range / azimuth coordinates follows the same processing sequence, the only difference being in the program that generates the initial geocoding lookup table. This is supported by the program ***gec_map_grd***.

Generation of ISP/SLC parameter file for the SAR image to be geocoded and orbits improvement, if possible, must be carried out before generating the image to be geocoded.

The Table below summarizes these steps highlighting the programs required

Step	**Program(s) used**
1. DEM/MAP parameter file creation	***create_dem_par***
2. Derivation of lookup table	***gec_map (gec_map_grd)***
3. Backward geocoding from SAR to map coordinates	***geocode_back***

C.1. Introduction

To demonstrate GEC geocoding we consider the ERS-1 image also considered in Example B for terrain corrected geocoding. The image was acquired over Las Vegas on 23 May 1996. As file identifier we use the orbit number, 25394. The image is in SLC format. Aim is geocoding the SAR intensity image obtained to UTM projection with 25 m posting.

The following files are used in the processing example. These files are available on the DEMO CD-ROM distributed with the software.

Filename	**Content**
25394.slc	SLC image
25394.slc.par	SLC parameter file (reference geometry)

Geocoding will consist of the following processing steps

- Generation of MLI intensity image from SLC image in case (see Section B.2)
- Generation of DEM/MAP parameter file
- Derivation of geocoding lookup table
- Geocoding (backward) of SAR intensity image to map geometry

C.2. Generation of DEM/MAP parameter file

To create the DEM parameter file the program ***create_dem_par*** must be used. The command line looks as follows

```
create_dem_par dv_utm.dem_par 25394.mli.par
```

The DEM parameter file describing the DEM in UTM projection is called dv_utm.dem_par. In this case we also use the parameter file of the intensity image to aid locating the DEM geographically.

Since this is an interactive program the user is asked to input some information to create the parameter file. Below we summarize the values to be selected. For more information on the creation of a DEM parameter file, refer to Example A, Section A.2.

```
- Select DEM projection -
DEM projection (EQA, UTM, OMCH, …): EQA : UTM

- Select datum-
enter country name: USA
number of available datums for country: USA : 5

ID:    1  DATUM: North American 1927   ELLIPSOID: Clarke 1866
…
ID:    5  DATUM: WGS 1984   ELLIPSOID: WGS 84 REGION: Global Definition, WGS84, World
select DATUM ID number in the range 1 - 5: 5

-Select projection-
input UTM projection parameters : proposed UTM zone number: 11
UTM zone number: (11): 11

- Selection of further parameters
DEM title: DEM : Nevada_srtm_utm
data format(REAL*4, INTEGER*2): REAL*4 : INTEGER*2
DEM height offset (m) (nominal=0.0):        0.000: 0.0
DEM height scale factor (nominal=1.0):    1.00000: 1.0
DEM width (samples):   1526: 4500
DEM length (lines):    1299: 4000
posting (northing (m), easting (m)):   -50.000000    50.000000 : -25.0  25.0
offset of first DEM sample (northing (m), easting (m)):  4036000.000   626000.000 : 4036000.000   626000.000
```

The size of the DEM is suggested by the program and is based on the default posting values (50 m in this case) and the SAR intensity image corner coordinates. Here we decided to use the values 4500 and 4000 because of the smaller pixel size and in order to be sure to include the whole area covered by the SAR image.

The provision of the SLC parameter file as input allows the programs to compute the corner coordinates of the DEM in the given projection based on the area covered by the SLC.

Errors in the DEM parameter file resulting from incorrect typing can be corrected by opening the DEM parameter file with a text editor. This also allows modifying one or more parameters in case the geocoding has not been successful (e.g. part of area covered by SAR image not included as a result of too small width/length or offset of first DEM sample).

C.3. Generation of geocoding lookup table

Once the DEM/MAP parameter file and the SAR image to be geocoded are ready the transformation between the radar and the map geometry can be established, i.e. the geocoding lookup table can be generated. To generate the lookup table the program ***gec_map*** must be used. The basic command line in this example looks as follows

```
gec_map    25394.mli.par    -    dv_utm.dem_par    0.0    LasVegas.dem_par
LasVegas.utm_to_rdc 1 1
```

In this example the program ***gc_map*** produces

- the DEM/MAP parameter file for the area covered by the SAR image (LasVegas.dem_par)
- the geocoding lookup table (LasVegas.rough.utm_to_rdc)

In this example we assumed the constant height value to be 0 m.

By looking at the DEM parameter file for the DEM segment we can see that the segment is fairly smaller than the initial DEM, being 2248 pixels wide and 1794 pixels long.

C.4. Backward geocoding from SAR to map coordinates

The SAR intensity image can now be geocoded from SAR range-Doppler geometry to UTM map geometry using the lookup table. The program for backward geocoding is ***geocode_back*** and in this example the command line looks as follows

```
geocode_back 25394.mli 2500 LasVegas.utm_to_rdc 25394.mli.utm 2248 0 2
```

Besides the input and output image file names and the lookup table, the other parameters required are the width of the image in radar geometry (2500) and the width of the image in map coordinates (2248). In this case it is set to be equal to the size of the lookup table. Various interpolation modes are supported. For a SAR intensity image in floating point format the spline-log (option 2) format is the most appropriate one. For SUN raster files only nearest-neighbor can be applied. In this case, attention has to be paid to the left/write flipping of the input data (usually left/right flipping is applied when a SUN/BMP raster file is created in radar geometry).

The geocoded image can be displayed with the DISP program ***dispwr:***

```
dispwr 25394.mli.utm 2248
```

Figure C1 shows the geocoded SAR intensity image. Compared to the terrain geocoded image in Figure B6, it is straightforward that the GEC image was not correctly geocoded in sloped areas.

If a map is available, the quality of the geocoded image can be determined. It is expected to found geocoding errors also for flat areas because of the quality of the orbital data. If necessary, a list of ground control points can be manually selected with ***gcp_ras*** or ***gcp_2ras*** and used to determine the fine registration polynomial with ***offset_list_fitm*** and refine the geocoding lookup table with ***gc_map_fine***.

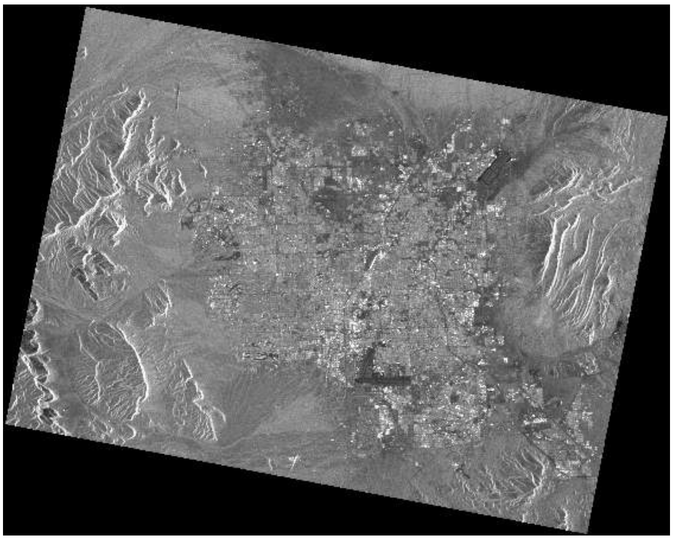


Figure C1. Ellipsoid geocoded SAR intensity image in UTM projection with 25 m posting

D. Geocoding with InSAR heights

For geocoding with INSAR heights an elevation map in slant range geometry obtained from an interferometric pair has to be available. In this case geocoding consists of the following steps:

- DEM/MAP parameter file creation using the program ***create_dem_par***
- Derivation of geocoding lookup table that links map geometry to range-Doppler geometry. This is done using the program ***gc_insar***.
- The resulting lookup table is used for the (forward) geocoding from SAR to map coordinates.

The Table below summarizes these steps highlighting the programs required

Step	**Program(s) used**
1. DEM/MAP parameter file creation	***create_dem_par***
2. Derivation of initial lookup table and SAR intensity image simulation	***gc_insar***
3. Forward geocoding from SAR to map coordinates	***geocode***

D.1. Introduction

To demonstrate the geocoding with an InSAR elevation map we consider the ERS-2 image acquired over Las Vegas on 24 May 1996 that has already been used for demonstrating interferometric processing in the User's Guide of the ISP module. As file identifier we use the orbit number, 05721. The image is in SLC format. Aim is geocoding the SAR intensity image to UTM projection with 25 m posting. As DEM we the elevation map obtained with the interferometric processing sequence described in the User's Guide of the ISP module.

The following files are used in the processing example. These files are available on the DEMO CD-ROM distributed with the software. The SLC image and the parameter file are available in the directory slc/25394. The elevation map in radar geometry and the ISP offset parameter file are available in the directory dem

Filename	**Content**
05721.slc	MLI
05721.slc.par	SLC parameter file (reference geometry)
05721_25394.hgt	Elevation map in radar geometry
05721_25394.off	ISP offset parameter file

Geocoding will consist of the following processing steps

- Generation of MLI intensity image from SLC image in case (see Section B.2)
- Generation of DEM/MAP parameter file
- Derivation of geocoding lookup table
- Geocoding (forward) of SAR intensity image to map geometry

D.2. Generation of DEM/MAP projection

At first it is necessary to create the DEM/MAP parameter file for the projection to be used. In this way the link between the radar and the map geometry can be established.

To create the DEM parameter file the program ***create_dem_par*** must be used. The command line looks as follows

```
create_dem_par dv_utm.dem_par 05721.mli.par
```

The DEM parameter file describing the DEM in UTM projection is called dv_utm.dem_par. In this case we also use the parameter file of the intensity image to aid locating the DEM geographically.

Since this is an interactive program the user is asked to input some information to create the parameter file. Below we summarize the values to be selected. For more information on the creation of a DEM parameter file, refer to Example A, Section A.2.

```
- Select DEM projection -
DEM projection (EQA, UTM, OMCH, …): EQA : UTM

- Select datum-
enter country name: USA
number of available datums for country: USA : 5

ID:    1  DATUM: North American 1927   ELLIPSOID: Clarke 1866
…
ID:    5  DATUM: WGS 1984   ELLIPSOID: WGS 84 REGION: Global Definition, WGS84, World
select DATUM ID number in the range 1 - 5: 5

-Select projection-
input UTM projection parameters : proposed UTM zone number: 11
UTM zone number: (11): 11

- Selection of further parameters
DEM title: DEM : Insar_dem
data format(REAL*4, INTEGER*2): REAL*4 : REAL*4
DEM height offset (m) (nominal=0.0):       0.000: 0.0
DEM height scale factor (nominal=1.0):    1.00000: 1.0
DEM width (samples):   1526: 3000
DEM length (lines):    1299: 2500
posting (northing (m), easting (m)):   -50.000000    50.000000 : -25.0  25.0
offset of first DEM sample (northing (m), easting (m)):  4036000.000   626000.000 : 4036000.000   626000.000
```

The size of the DEM in the new projection is suggested by the program and is based on the default posting values (50 m in this case) and the SAR intensity image corner coordinates. Here we decided to use the values 2500 and 2000 because of the smaller pixel size and in order to be sure to include the whole area covered by the SAR image.

The provision of the SLC parameter file as input allows the programs to compute the corner coordinates of the area to be considered in the projection.

Errors in the DEM parameter file resulting from incorrect typing can be corrected by opening the DEM parameter file with a text editor. This also allows modifying one or more parameters in case geocoding has not been successful (e.g. part of area covered by SAR image not included as a result of too small width/length or offset of first DEM sample).

D.3. Generation of geocoding lookup table

Once the DEM parameter file and the SAR image to be geocoded are ready the transformation between the radar and the map geometry can be established, i.e. the geocoding lookup table can be generated. To generate the lookup table based on elevation map in radar geometry the program ***gc_insar*** must be used. The command line in this example looks as follows

```
gc_insar 05721.slc.par 05721_25394.off 05721_25394.hgt dv_utm.dem_par
LasVegas.rdc_to_utm
```

The geocoding lookup table (LasVegas.rdc_to_utm) links the radar geometry (reference) with the map geometry. The radar geometry is described in the SLC parameter file. The DEM geometry is described in the ISP offset parameter file. The lookup table can be displayed with the DISP program ***dismph***, taking into account that the size of the lookup table coincides with the one of the SAR image:

```
dismph LasVegas.rdc_to_utm 2500
```

The accuracy of the geocoding lookup table is mainly limited by the accuracy of the available orbit geometry. No automated refinement of the geocoding lookup table is foreseen.

D.4. Forward geocoding from SAR to map coordinates

The lookup table can now be used to transform files from the geometry in which the lookup table is given (radar) to the linked geometry (map). This is why we refer to this geocoding as forward transformation. Geocoding of SAR images (and the height map) is therefore done with the program ***geocode***. For example the command line for transforming the MLI image to UTM projection is the following:

```
geocode LasVegas.rdc_to_utm 05721.mli 2500 05721.mli.utm 3000 - 1 0
```

The width of the SAR intensity image in map projection can be found in the DEM parameter file (3000). The image can be displayed with the DISP program ***dispwr***:

```
dispwr 05721.mli.utm 3000
```

Figure D1 shows the geocoded image. The SAR image has not been geocoded in areas that were not unwrapped when generating the elevation map because of layover/shadow effects.

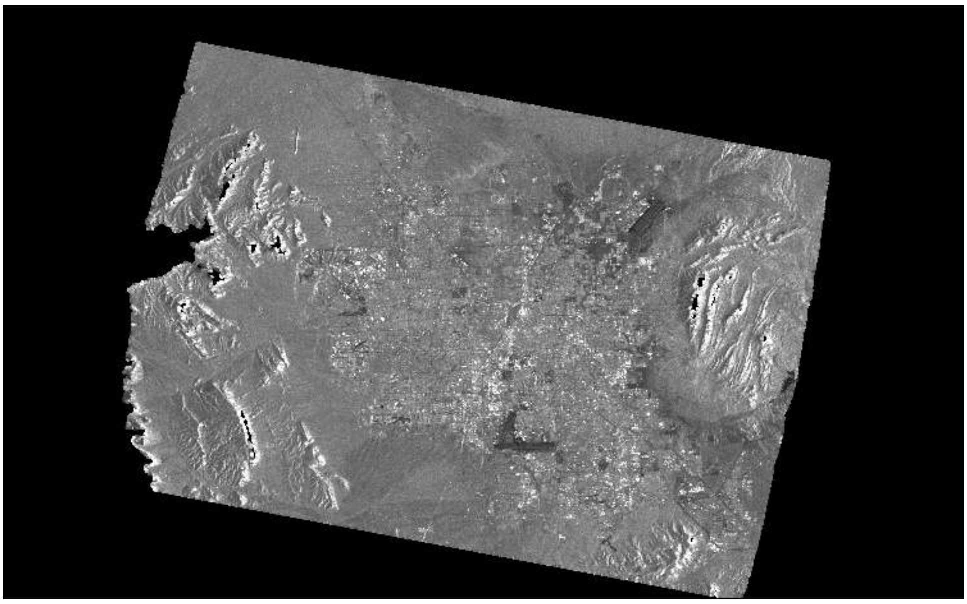


Figure D1. SAR intensity image geocoded to UTM projection, 25-m posting, using an InSAR-derived elevation map.

E. Multi-source image registration – Example with Landsat image

The following example illustrates a typical processing sequence for image registration of SAR and optical satellite images of similar resolution. The satellite images are assumed to be already geocoded using a geocoding lookup table. If both images have been geocoded with high accuracy the registration of one image with respect to the other would just need the program ***map_trans***. However, not perfect matching of geocoded satellite images is usually observed because of inaccuracies in the orbital data. For this reason the co-registration requires a refinement process that can correct for inaccuracies in one or both geocoded products.

A similar processing sequence may be also applied to geocoded SAR images of different frequencies, incidence angles and orbits (e.g. ERS and JERS, ENVISAT ASAR data of different swaths).

For multi-source image registration two geocoded images in map coordinates are needed. It is not relevant if they are in the same projection and/or have the same pixel size. In this case the following steps are performed:

- Transformation of one dataset to the geometry of the image considered as reference and generation of the lookup table describing the transformation. If one or both images present errors due to inaccurate geocoding, this will reflect in the lookup table which will not be fully accurate.
- Fine registration of the transformed image with the image considered as reference (***create_diff_par*** followed by ***init_offsetm***, ***offset_pwrm***, and ***offset_fitm***). The offset registration polynomials in the DIFF parameter file are then used to refine the lookup table (***interp_cpx***).
- The resulting refined lookup table is used for the (backward) transformation of the image to be resampled to the reference image (***geocode_back***).

The Table below summarizes these steps highlighting the programs required

Step	**Program(s) used**
1. Transformation to the same projection and pixel spacing / generation of transformation lookup table	***map_trans***
2. Co-registration between image to be resampled and reference image	***create_diff_par, init_offsetm, offset_pwrm, offset_fitm***
3. Refinement of lookup table	***interp_cpx***
4. Resampling	***geocode_back***

E.1. Introduction

To demonstrate multi-source image registration we consider the terrain geocoded ERS-1 SAR image described in Example B, which was acquired over Las Vegas on 23 May 1996 and the panchromatic channel of a Landsat ETM+ image acquired on 3 May 2000. The Landsat is in UTM (zone 11) projection with -14.25 m pixel size. In this example we will show how to co-register the Landsat image to the SAR image, which is in UTM projection with 25 m posting to the geocoded SAR image is assumed to be the reference.

If the optical image had been the reference and we would have wanted to co-register the geocoded SAR image with the optical image, the SAR image takes the place of the optical image and vice versa in the steps described from Sections E.3 onward

The files used in the processing example are listed in the table below. These files are not available on the DEMO CD-ROM distributed with the software. The SAR geocoded image can be obtained from processing the DEMO-CD data as described in Example B. The Landsat image can be downloaded from the Earth Science Data Interface (ESDI) at the Global Land Cover Facility (GLCF) website (path 39, row 35). The Geo-cover image is in tiff format.

Filename	**Content**
25394.mli.utm	Geocoded SAR image (width: 2376, length: 1848)
LasVegas.dem_par	DEM parameter file describing geometry of the geocoded SAR image
p039r035_7p20000503_z11_nn80.tif	Landsat image

The registration will consist of the following processing steps

- Generation of DEM parameter file for Landsat image (if not available yet)
- Transformation of Landsat image to the geometry of the geocoded SAR image and derivation of lookup table for the transformation
- Offset computation between the transformed Landsat image and the SAR intensity image
- Refinement of transformation lookup table
- Resampling of Landsat image to the geometry of the geocoded SAR image

E.2. Generation of DEM/MAP parameter file for Landsat image

For the registration it is necessary that both images have their own DEM/MAP parameter file. In this example the DEM/MAP parameter file of the geocoded SAR image is already available since it has been obtained in Example B (LasVegas.dem_par).

For the Landsat image parameters are taken from the metadata information delivered with the Landsat image. To obtain from the tiff file the raster image and a separate file containing metadata information necessary for the generation of the DEM parameter file the tool gdal_translate of the open source package GDAL (not part of the GAMMA Software) can be used as follows:

```
gdal_translate -of ENVI p039r035_7p20000503_z11_nn80.tif LSAT
```

where LSAT is the file name of the raster image. The metadata information is stored in the *.hdr text file generated by gdal_translate.

The Landsat image is in byte format, hence to display it the program ***disbyte*** shall be used. To display a subset of 2000 lines starting at line 10000 the following command can be used

```
disbyte LSAT 17310 10000 2000
```

The DEM/MAP parameter file describing the Landsat image and its projection parameters can be generated with the program ***create_dem_par*** as follows:

```
create_dem_par LSAT.dem_par
```

The LSAT.dem_par file is shown below

```
Gamma DIFF&GEO DEM/MAP parameter file
title: Landsat_Geocover
DEM_projection:     UTM
data_format:        REAL*4
DEM_hgt_offset:          0.00000
DEM_scale:               1.00000
width:              17310
nlines:             15300
corner_north:  4100672.625   m
corner_east:    571845.375   m
post_north:    -14.2500000   m
post_east:      14.2500000   m

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

projection_name: UTM
projection_zone:                 11
false_easting:           500000.000   m
false_northing:               0.000   m
projection_k0:            0.9996000
center_longitude:      -117.0000000   decimal degrees
center_latitude:          0.0000000   decimal degrees
```

E.3. Transformation of Landsat image to same projection and pixel spacing of SAR image

The transformation of the Landsat image to the SAR map projection is performed with the program ***map_trans***. Here we assume that the Landsat image is simply called LSAT.

```
map_trans LSAT.dem_par LSAT LasVegas.dem_par LSAT.utm_sar - - 0 3 SAR_to_LSAT.rough
```

The lookup table for the transformation called SAR_to_LSAT.rough describes the transformation between the geometries of the two images. If one or both images present geocoding errors the transformation in the lookup table is not fully correct.

The SAR and Landsat images 25394.utm.mli and LSAT.utm_sar are in the same map projection with the same posting. In order to work further with the Landsat image a conversion from byte to floating point format is necessary. This can be done with the program ***uchar2float***. The command line looks as follows:

```
uchar2float LSAT.utm_sar LSAT.utm_sar.float
```

The SAR and the Landsat image can now be visualized with the DISP program ***dis2pwr***:

```
dis2pwr 25394.mli.utm LSAT.utm_sar.float 2376 2376
```

Although the two images appear to almost perfectly overlap, there is still a residual offset. The not perfect matching is observed because of inaccuracies in the orbital data. For this reason a refinement of the transformation is required. This is done by computing offsets between the two images and using these offsets to construct a polynomial model that refines the lookup table for the transformation.

E.4. Co-registration and lookup table refinement

Offsets between the two images are determined by first creating a DIFF_par parameter file that will describe the offsets between the transformed Landsat and the geocoded SAR image:

```
create_diff_par LasVegas.dem_par - 25394_LSAT.diff_par 2
```

At first we determine a global offset between the two images by using the program ***init_offsetm*** as follows

```
init_offsetm 25394.mli.utm LSAT.utm_sar.float 25394_LSAT.diff_par
```

The initial offsets estimate has a SNR value below the given threshold and for this reason it rejected.

Accurate offsets are then determined for a large number of small image chips (by optimizing the cross-correlation of the SAR and the optical images). This is done with the program ***offset_pwrm*** as follows

```
offset_pwrm 25394.mli.utm LSAT.utm_sar.float 25394_LSAT.diff_par offs snr
128 128 offsets 1 16 16
```

The offsets estimates are then used to determine the bi-linear fine registration polynomial. This is done with the program ***offset_fitm*** which in this example is used as follows:

```
offset_fitm offs snr 25394_LSAT.diff_par coffs coffsets 7.0 3
```

The sequence ***offset_pwrm / offset_fitm*** can be repeated a couple of times to improve the quality of the estimate. The bilinear co-registration function is used to guide the search of the positions where cross-correlation is applied. Typically an improvement of the estimates is noticed when size of the windows is adapted to the size of the matching features in the two images and the number of windows is such that all matching features are covered. The order of the polynomial can be increased if the co-registration appears poor.

Instead of this automated method, a list of control points can be manually selected with ***gcp_2ras*** and used to determine the fine registration polynomial with ***offset_list_fitm***.

In this case we run the offset estimation sequence using more windows (24x24 instead of 16x16)

```
offset_pwrm 25394.mli.utm LSAT.utm_sar.float 25394_LSAT.diff_par offs snr 128 128 offsets 1 24 24
```

```
offset_fitm offs snr 25394_LSAT.diff_par coffs coffsets 7.0 3
```

It is important to remind that number of windows and size depends on the specific example being run.

The bilinear fine registration polynomial stored in the DIFF_par parameter file is then used to refine the lookup table. This is done with the program ***interp_cpx***:

```
interp_cpx SAR_to_LSAT.rough 25394_LSAT.diff_par SAR_to_LSAT 0 - 1
```

Here we used a bilinear spline for the interpolation. This is a suitable interpolation mode for this example.

E.5. Resampling of Landsat image to match with SAR image

With the refined lookup table, SAR_to_LSAT, resampling from the original Landsat geometry of the image LSAT to the map geometry of the SAR image can be done in a single step. The program to be used is ***geocode_back***. In our example the command line looks as follows:

```
geocode_back LSAT 17310 SAR_to_LSAT LSAT.utm_sar 2376 1848 0 3
```

The image LSAT.utm_sar is now replaced with a new version based on the refined transformation between the geometries of the geocoded SAR image and the Landsat image.

The refined lookup table is also useful for other channels of the optical image.

To display the geocoded SAR image and the refined Landsat image simultaneously, first a byte to floating point conversion is needed after which the DISP program ***dis2pwr*** can be used.

```
uchar2float LSAT.utm_sar LSAT.utm_sar.float
```

```
dis2pwr 25394.mli.utm LSAT.utm_sar.float 2376 2376
```

F. Multi-source image registration – Example with ASTER image in SUNraster format

While the previous example showed how to co-register a geocoded Landsat image in binary format to a geocoded SAR image, this example deals with the image registration of images in SUNraster format.

The optical image to be registered to the SAR image is an ASTER image (Level-1B, Band 2, 0.63-0.69 micro m). The image is given as an 8-bit SUNraster file (255 bits). It is given in UTM projection, zone 11 at 15 m spatial resolution (i.e. a rotation of the original Level-1B data in hdf format was applied).

The geocoded ASTER image will be co-registered to the geocoded ERS-1 Multi-Look Intensity (MLI) image. The geocoded MLI (also in UTM projection, zone 11, but at 25 m spatial resolution) is the output of Example B (and of the DEMO script run_GEO_LasVegas).

The processing sequence consists of:

- Transformation from SUNraster to float format
- Transformation of the ASTER image to the map geometry of the SAR image considered as reference and generation of the lookup table describing the transformation. If one or both images present errors due to inaccurate geocoding, this will reflect in the lookup table which will not be fully accurate.
- Fine registration of the transformed image with the image considered as reference (***create_diff_par*** followed by ***init_offsetm***, ***offset_pwrm***, and ***offset_fitm***). The offset registration polynomials in the DIFF parameter file are then used to refine the lookup table (***interp_cpx***).
- The resulting refined lookup table is used for the (backward) transformation of the image to be resampled to the reference image (***geocode_back***).

The Table below summarizes these steps highlighting the programs required

Step	**Program(s) used**
1. Transformation to of SUNraster image to floating point	***ras_linear***
2. Transformation to the same projection and pixel spacing / generation of transformation lookup table	***map_trans***
3. Co-registration between image to be resampled and reference image	***create_diff_par, init_offsetm, offset_pwrm, offset_fitm***
4. Refinement of lookup table	***interp_cpx***
5. Resampling	***geocode_back***

F.1. Introduction

The files used in the processing example are listed in the table below. The files for the SAR image have been obtained in Example B, hence Example B has to be run first if not done yet. The ASTER files are available on the DEMO CD-ROM in the directory aster.

Filename	Content
25394.mli.utm	Geocoded SAR image (width: 2376, length: 1848)
LasVegas.dem_par	DEM parameter file describing geometry of the geocoded SAR image
LasVegas.aster.2.utm.ras	ASTER image in SUNraster format (width: 5620)
LasVegas.aster.2.utm.dem_par	DEM parameter file for ASTER image

The registration will consist of the following processing steps

- Transformation of ASTER image to the geometry of the geocoded SAR image and derivation of lookup table for the transformation
- Offset computation between the transformed ASTER image and the SAR intensity image
- Refinement of transformation lookup table
- Resampling of ASTER image to the geometry of the geocoded SAR image

The processing is also supported by an automated processing script in the DEMO-CD scripts directory (run_IR_LasVegas). The list of commands in this script can be found in the file com_IR_LasVegas.

The script should be considered as an introduction to scripting and can be used for developing own scripts based on the user's particular needs. If the script is used for processing, it is strongly recommended to adapt it by selecting the programs actually needed for processing and by critically choosing the values of the parameters required by each individual program. For this purpose it is highly recommended to refer to the Reference Guide.

F.2. Conversion of image from SUNraster to float format

In order to proceed with the co-registration the two datasets have to be in float format. This requires that the ASTER image is converted to floating point format. This is done with the DISP program ras_linear. In this example the command line looks as follows:

```
ras_linear LasVegas.aster.2.utm 5620 1 0 1 1 0. 255. 1
LasVegas.aster.2.utm.ras -1
```

F.3. Transformation of ASTER image to same projection and pixel spacing of SAR image

The transformation of the ASTER image to the SAR map projection is performed with the program ***map_trans***. Here we assume that the Landsat image is simply called LSAT.

```
map_trans LasVegas.aster.2.utm.dem_par LasVegas.aster.2.utm
LasVegas.dem_par LasVegas.aster.2.utm_rough_sar 1 1 0 0
LasVegas.sar_to_aster_rough
```

The lookup table for the transformation called LasVegas.sar_to_aster.rough describes the transformation between the geometries of the two images. If one or both images present geocoding errors the transformation contained in the lookup table is not fully correct.

The geocoded SAR image, 25394.utm.mli, and the geocoded ASTER image, LasVegas.aster.2.utm_rough_sar, are in the same map projection with the same posting.

The two images can be displayed with the DISP program ***dis2pwr***:

```
dis2pwr 25394.mli.utm LasVegas.aster.2.utm_rough_sar 2376 2376
```

Although the two images appear to almost perfectly overlap, there is still a residual offset. The not perfect matching is observed because of inaccuracies in the orbital data. For this reason a refinement of the transformation is required. This is done by computing offsets between the two images and using these offsets to construct a polynomial model that refines the lookup table for the transformation.

F.4. Co-registration and lookup table refinement

Offsets between the two images are determined by first creating a DIFF_par parameter file that will describe the offsets between the transformed Landsat and the geocoded SAR image:

```
create_diff_par LasVegas.dem_par - LasVegas.diff_par 2
```

Since the offsets between the images are very small we can skip the computation of the initial offsets estimates and determine accurate offsets for a large number of small image chips (by optimizing the cross-correlation of the SAR and the optical images). This is done with the program ***offset_pwrm*** as follows

```
offset_pwrm 25394.mli.utm LasVegas.aster.2.utm_rough_sar LasVegas.diff_par
offs snr 256 256 offsets 2 8 8 7.0
```

The offsets estimates are then used to determine the bi-linear fine registration polynomial. This is done with the program ***offset_fitm*** which in this example is used as follows:

```
offset_fitm offs snr LasVegas.diff_par coffs coffsets 7.0 3
```

To improve the estimates of the coefficients of the polynomial the sequence offset_pwrm / offset_fitm is run one more time. This tume the offset estimation sequence will make use of more windows (24x24 instead of 8x8)

```
offset_pwrm 25394.mli.utm LasVegas.aster.2.utm_rough_sar LasVegas.diff_par
offs snr 256 256 offsets 2 24 24 7.0

offset_fitm offs snr LasVegas.diff_par coffs coffsets 7.0 3
```

It is important to remind that number of windows and size depends on the specific example being run.

The bilinear fine registration polynomial stored in the DIFF_par parameter file is then used to refine the lookup table. This is done with the program ***interp_cpx***:

```
interp_cpx LasVegas.sar_to_aster_rough LasVegas.diff_par LasVegas.sar_to_aster 1 - 1
```

Here we used a bilinear spline for the interpolation. This is a suitable interpolation mode for this example.

F.5. Resampling of ASTER image to match with SAR image

With the refined lookup table, LasVegas.sar_to_aster, resampling from the original ASTER geometry of the image to the map geometry of the SAR image can be done in a single step. The program to be used is ***geocode_back***. In our example the command line looks as follows:

```
geocode_back LasVegas.aster.2.utm 5620 LasVegas.sar_to_aster LasVegas.aster.2.utm_sar 2376 1848 2 0
```

The image LasVegas.aster.2.utm_sar represents the co-registered version of the ASTER image to the SAR image .

To display the geocoded SAR image and the co-registered ASTER image simultaneously, the DISP program ***dis2pwr*** can be used.

```
dis2pwr 25394.mli.utm LasVegas.aster.2.utm_sar 2376 2376
```

G. Geocoding using an external reference image

In Example B we showed how to do terrain geocoding with a DEM. The procedure is very robust and accurate when there are enough offsets to generate a reliable offset polynomial. This requires enough windows in which reliable offsets can be found. This in turn requires the simulated SAR image to present some sort of texture, primarily related to topographic relief, and that this is also visible in the SAR intensity image. If this is not the case, the refinement of the geocoding lookup table will not be entirely correct and residual errors will remain.

Poor offset estimation occurs in areas of moderate to no topography or for sensors that have shallow incidence angles thus being less sensitive to topography (e.g. PALSAR or JERS).

In such cases one possibility is to define an initial geocoding lookup table from orbital data and then refine it using the same co-registration procedure between the SAR image and simulated SAR image from the DEM, this time using an accurately geocoded image (instead of the simulated SAR image). The external reference image can be for example an optical image. In this case the optical image has to be put first in the geometry of the map reference system to which the SAR image will be geocoded.

The Table below summarizes the processing steps highlighting the programs required

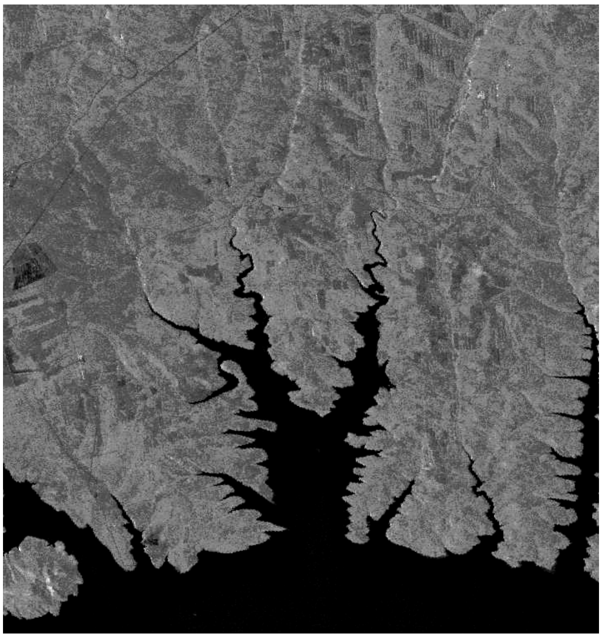
Step	**Program(s) used**
1. DEM/MAP parameter file creation for reference DEM (if available)	***create_dem_par***
2. Derivation of initial geocoding lookup table	***gc_map (gc_map_grd)***
3. Generation of DEM/MAP parameter file for external reference image	***create_dem_par***
4. Transformation of external reference image to the map geometry to which SAR image will be geocoded	***map_trans***
5. Transformation of external reference image from map to radar geometry	***geocode***
6. Fine registration between external reference image and SAR image	***create_diff_par, init_offsetm, offset_pwrm, offset_fitm***
7. Refinement of initial geocoding lookup table	***Gc_map_fine***
8. Backward geocoding from SAR to map coordinates	***geocode_back, geocode***

In this Section we show an example for a PALSAR image for which the standard deviation of offsets after refinement were larger than 1 pixel in range and 2 pixels in azimuth. This large error was caused primarily by the absence of topographic features. Aim of this example is to geocode the SAR intensity image to an Albers Conical Equal Area projection with 50 m posting using an external reference image.

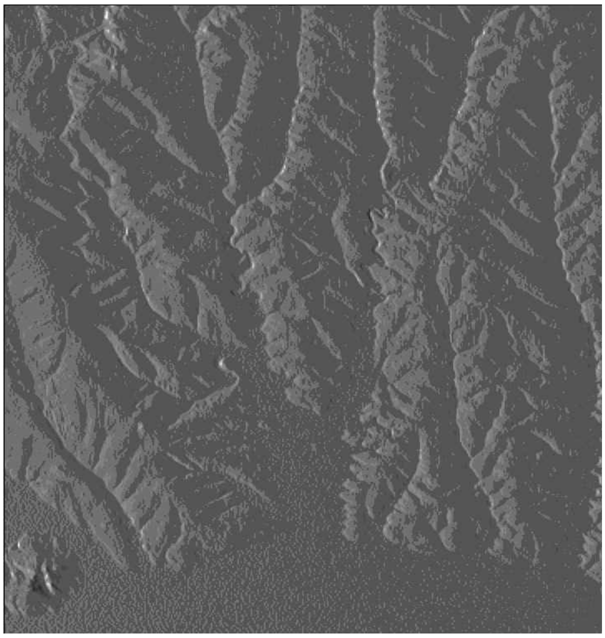
The SAR image is called 200708_hh.mli and is in MLI format. It is 7000 pixels long and 1248 pixels wide with approximately 50 m pixel spacing in (ground) range and azimuth.

As external reference image we will use a Landsat ETM+ panchromatic image in UTM projection with -14.25 m pixel spacing downloaded from the Global Land Cover Facility (GLCF).

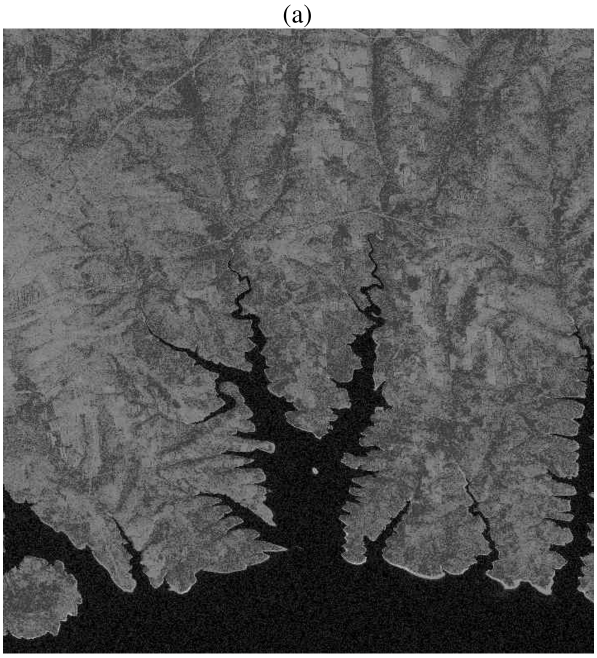
Figure G1 shows that the SAR intensity image has many more features in common with the Landsat image than with the simulated SAR image. This is due to the weak sensitivity of PALSAR backscatter to topography.



(a)



(b)



(c)

(a) PALSAR intensity image

(b) Simulated SAR image from DEM

(c) Landsat panchromatic image

Figure G1. Comparison between PALSAR intensity image, simulated SAR image from DEM and Landsat panchromatic image for a subset of the area considered in this example.

The files required in the processing example are listed in the table below. These files are not available on the DEMO CD-ROM distributed with the software. The commands listed below can be easily adapted to an own dataset.

Filename	Content
20070819_hh.mli	PALSAR MLI SAR intensity image (HH-pol)
20070819_hh.mli.par	ISP SLC/MLI parameter file of MLI image
dem	DEM used to generate lookup table
dem_par	DEM parameter file describing geometry of the geocoded SAR image
p039r035_7p20000503_z11_nn80.tif	Landsat image

The geocoding will consist of the following processing steps

- Generation of DEM parameter file for reference DEM (see Example A and B for details)
- Derivation of initial geocoding lookup table
- Generation of DEM parameter file for Landsat image (if not available yet)
- Transformation of Landsat image to the geometry of the geocoded SAR image
- Using the initial geocoding lookup table, transformation of Landsat image from map to radar geometry
- Offset computation between the Landsat image in radar geometry and the SAR intensity image
- Refinement of geocoding lookup table
- Resampling of SAR intensity image to map geometry

G.1 Derivation of initial lookup table

In this example we assume that a DEM and corresponding DEM parameter file are already available (dem and dem_par). For information on how to generated a DEM parameter file, refer to Example A and B.

The initial geocoding lookup table is obtained with the program ***gc_map***. This program will also define the parameters of the geometry to which the SAR image shall be geocoded.
The command line for ***gc_map*** looks as follows:

```
gc_map 20070819_hh.mli.par - dem.par dem dem_seg.par dem_seg
map_to_rdc.rough 1 1 sim
```

The files dem and dem.par represent the DEM used for generating the initial lookup table and the corresponding DEM parameter file. The program generates the DEM segment and the corresponding DEM parameter file (dem_seg and dem_seg.par). The DEM parameter file for the segment contains all the parameters for the geocoded SAR product. The box below contains the DEM parameter file of the DEM segment.

With the processing sequence presented in Example B, the geocoded SAR image was affected by non-negligible offsets in both northing and easting direction. This is due to the rather flat topography and the absence of clear features in the simulated SAR intensity image obtained from the DEM. The PALSAR image has instead many features that can also be observed in the Landsat image. For this reason it can be foreseen that the automatic registration procedure applied to the Landsat image instead of the simulated SAR image will perform with greater accuracy.

```
Gamma DIFF&GEO DEM/MAP parameter file
title: DEM
DEM_projection:     AEAC
data_format:        REAL*4
DEM_hgt_offset:          0.00000
DEM_scale:               1.00000
width:              3344
nlines:             7430
corner_north:   924650.000   m
corner_east:   1136250.000   m
post_north:    -50.0000000   m
post_east:      50.0000000   m
first_std_parallel:      56.000000   decimal degrees
second_std_parallel:     73.000000   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

projection_name: AEAC
projection_zone:                 0
false_easting:         1000000.000   m
false_northing:              0.000   m
projection_k0:           1.0000000
center_longitude:       99.0000000   decimal degrees
center_latitude:        50.0000000   decimal degrees
```

G.2. Generation of DEM/MAP parameter file for external reference image

Depending on the external reference image, it is not only necessary to generate a DEM/MAP parameter file but also to prepare the image so that it can be treated by the GAMMA Software.

The Landsat image here considered is in tiff format, which requires stripping off the header to obtain a simple raster image as required by the GAMMA Software. The image file can be obtained for example using the open source GDAL program gdal_translate (this is not part of the GAMMA Software). A file *.hdr is also generated in which all parameters required for the generation of the DEM/MAP parameter file are listed. This is created with the program ***create_dem_par*** as described below. The command lines are listed below. The DEM/MAP parameter file for the Landsat image is in the box below.

```
gdal_translate –of ENVI p136r021_7p20020821_z48_nn80.tif lsat
```

```
create_dem_par lsat.dem_par
```

The Landsat image has been saved to the file lsat. This file is in byte format.

```
Gamma DIFF&GEO DEM/MAP parameter file
title: Landsat_geotif
DEM_projection:     UTM
data_format:        REAL*4
DEM_hgt_offset:          0.00000
DEM_scale:               1.00000
width:              18528
nlines:             17030
corner_north:  6320024.625   m
corner_east:    238737.375   m
post_north:    -14.2500000   m
post_east:      14.2500000   m

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

projection_name: UTM
projection_zone:                48
false_easting:           500000.000   m
false_northing:               0.000   m
projection_k0:            0.9996000
center_longitude:      105.0000000   decimal degrees
center_latitude:         0.0000000   decimal degrees
```

G.3. Transformation of external reference from own to geocoding map projection

In order to proceed with the co-registration, the Landsat image has to be resampled to the map projection of the SAR image. In this way it will have the same coordinates as the map geometry to which the SAR image has to be geocoded. The transformation is done with the program ***map_trans***:

```
map_trans lsat.dem_par lsat dem_seg.par lsat.aeac_sar 1 1 0 3
```

Since the Landsat image is in byte format it is necessary to convert it to floating point format before starting with the co-registration. This is done with the program ***uchar2float:***

```
uchar2float lsat.aeac_sar lsat.aeac_sar.float
```

The image lsat.aeac_sar represents the reference image in the output map projection. It corresponds to the simulated SAR intensity image file LasVegas.sim_sar in Example B.

G.4. Co-registration

To determine the offsets between the Landsat image and the SAR image, the Landsat image has to be transformed from map to radar geometry. This is done with the program ***geocode***, in which the initial lookup table is used

```
geocode map_to_rdc.rough lsat.aeac_sar.float 3344 lsat.sar 1248 7000
```

The image lsat.sar represents the Landsat image in radar geometry. This image and the SAR image that needs to be geocoded can be displayed with the program ***dis2pwr:***

```
dis2pwr 20070819_hh.mli lsat.sar 1248 1248
```

The two images present clear offsets. To determine them the refinement procedure already described in Example B is used. At first a DIFF_par parameter file that will contain the offset information and the offset polynomials is initialized. This is done with the program ***create_diff_par***:

```
create_diff_par 20070819_hh.mli.par - diff_par 1
```

Then the initial offsets estimate is computed with the program ***init_offsetm***

```
init_offsetm 20070819_hh.mli lsat.sar diff_par
```

To determine the local offsets on which the coefficients of the offsets polynomial will be based the sequence ***offset_pwrm/offset_fitm*** is used. The sequence can be repeated as many time as required to obtain a reasonable offset polynomial. The reliability of the offset polynomial is indicated by the number of offsets used to estimate the polynomial coefficients and the standard deviations of the offsets. To improve results it is recommended using an increasing number of windows with decreasing size. However the size should not be too small in order to avoid missing relevant features within the estimation window.

For this example the sequence has been repeated several times. Here we also show how the estimates improved with increasing window number, decreasing window size and increasing number of polynomial coefficients. To keep all commands on one line we did not consider saving the offset estimates to a text file.

```
offset_pwrm 20070819_hh.mli lsat.sar diff_par offs snr 128 128 - - 8 32

offset_fitm offs snr diff_par coffs coffsets - 1
```

#final solution: 24 offset estimates accepted out of 256 samples
#final model fit std. dev. (samples) range: 0.5597 azimuth: 12.4776

```
offset_pwrm 20070819_hh.mli lsat.sar diff_par offs snr 128 128 - - 16 64

offset_fitm offs snr diff_par coffs coffsets - 1
```

#final solution: 155 offset estimates accepted out of 1024 samples
#final model fit std. dev. (samples) range: 0.4743 azimuth: 12.6914

```
offset_pwrm 20070819_hh.mli lsat.sar diff_par offs snr 128 128 - - 16 64

offset_fitm offs snr diff_par coffs coffsets - 3
```

#final solution: 151 offset estimates accepted out of 1024 samples
#final model fit std. dev. (samples) range: 0.3369 azimuth: 1.0554

```
offset_pwrm 20070819_hh.mli lsat.sar diff_par offs snr 128 128 - - 24 96
```

```
offset_fitm offs snr diff_par coffs coffsets - 4
```

#final solution: 333 offset estimates accepted out of 2304 samples
#final model fit std. dev. (samples) range: 0.2825 azimuth: 0.9526

```
offset_pwrm 20070819_hh.mli lsat.sar diff_par offs snr 96 96 - - 24 96
```

```
offset_fitm offs snr diff_par coffs coffsets - 4
```

#final solution: 213 offset estimates accepted out of 2304 samples
#final model fit std. dev. (samples) range: 0.2805 azimuth: 0.6806

```
offset_pwrm 20070819_hh.mli lsat.sar diff_par offs snr 96 96 - - 32 96
```

```
offset_fitm offs snr diff_par coffs coffsets - 4
```

#final solution: 310 offset estimates accepted out of 3072 samples
#final model fit std. dev. (samples) range: 0.2795 azimuth: 0.7018

The measure of the error in the refinement is below 1 pixel in both directions, thus being much better than when co-registering to the simulated SAR intensity image.

G.5. Refinement of geocoding lookup table

The refinement of the geocoding lookup table with the offset polynomial stored in the diff_par file is done with the program ***gc_map_fine***:

```
gc_map_fine map_to_rdc.rough 3344 diff_par map_to_rdc 0
```

To check whether the refinement was successful the Landsat image can be re-transformed with the refined lookup table to radar geometry:

```
geocode map_to_rdc lsat.aeac_sar.float 3344 lsat.sar2 1248 7000
```

Since the overlap in this example proved to be good, the refined lookup table map_to_rdc can now be used to geocoded the SAR intensity image.

G.6. Geocoding of SAR intensity image

To geocode the SAR intensity image we use the program ***geocode_back***:

```
geocode_back 20070819_hh.mli 1248 map_to_rdc 20070819_hh.geo.mli 3344
```

H. Measuring range and azimuth offsets on a regular grid in map geometry

It is possible using the DIFF/GEO software to measure range and azimuth offsets on a regular grid in a map geometry defined by a DEM parameter file using the programs ***dem_RDC_list*** and ***offset_pwr_list***. The offset locations can be plotted on a SUNraster file for reference, using program ***ras_clist***.

In the following example, a pair of RADARSAT-1 images acquired over a glacier in Antarctica is considered. The images were acquired on 19970924 and 19971018 (24 days apart). The starting point are the two images in SLC format that have been co-registered using the co-registration procedure described in the User's Guide of the ISP module. The images in SLC format images are 6638 range samples by 2500 azimuth lines with a pixel spacing of 8.11m slant range and a line spacing of 5.33m in azimuth.

Input to the processing sequence described in this example includes the co-registered SLC images (19970924_seg.slc and 19971018_seg.slc) and corresponding image parameter files, a 4x4 multilook intensity image of the reference image (19970924_seg.mli) and the associated parameter file also in radar coordinates. The MLI image is 1659 pixels wide and 625 pixels long.

At first both images are geocoded to the ellipsoid. Then a set of locations on regular grid in the map (in this case UTM) geometry is selected and transformed to the range-Doppler coordinates of the SLC. Range and azimuth offsets as well as the offset SNR are determined at these locations and stored as both a text list and as binary raster files.

The first step in this process requires ellipsoid geocoding of the SAR data to generate a geocoding lookup table. For this a DEM parameter file is necessary. This is generated with the program ***create_dem_par*** as follows

```
create_dem_par dem_par 19970924_seg.slc.par
```

The dem_par file is obtained aiding the program with the SLC parameter file of the reference image. To generate the geocoding lookup table based on ellipsoid geocoding the program ***gec_map*** is used as follows. The DEM parameter file of the geocoded image to be generated will be called dem_seg_par. The geocoding lookup table is called 19970924_0.map_to_rdc.

```
gec_map 19970924_seg.mli.par - dem_par 100 dem_seg_par 19970924_0.map_to_rdc 2.5 2.5
```

The DEM segment is 2896 pixels wide and 1262 pixels long.

We transform now the MLI image of the reference SLC from radar to map coordinates with the program ***geocode_back*** as follows:

```
geocode_back 19970924_seg.mli 1659 19970924_0.map_to_rdc 19970924_gec.mli 2896 1262 1 0
```

From the geocoded SAR image it is possible to generate a SUNraster image with the DISP program ***raspwr*** and display it with coordinates with the program ***disras_dem_par***:

```
raspwr 19970924_gec.mli 2896 1 0 1 1
```

```
disras_dem_par 19970924_gec.mli.ras dem_seg_par
```

The next step is to generate the list of locations in the geocoded image where the offsets are to be estimated. Offsets should only be measured at points in the map geometry with image coverage. Furthermore, the user may want to restrict offset estimation to a smaller patch in the geocoded image. The program that generates the list of coordinate pairs where to measure offsets is ***dem_RDC_list*** which can use a raster image that serves as a mask. This mask must be an 8-bit SUNraster or BMP image. If the value of the color table entry associated with the pixel is other than black (RGB=0,0,0), it is included in the regions where offsets can be measured.

The command line with ***dem_RDC_list*** in this example looks as follows:

```
dem_RDC_list dem_seg_par 19970924_0.map_to_rdc 19970924_seg.mli.par
19970924_gec.mli.mask.ras clist_RDC clist_MAP offset.dem_par 4 4
```

The coordinate lists clist_RDC and clist_MAP are text files consisting of pairs of numbers denoting the column and row coordinates.

The next step is to generate an image of the points in the geocoded MLI raster image using ***ras_clist*** to plot the patch center coordinates. The same points can be plotted in the RDC image using the clist_RDC values and the SUNraster file created from the MLI image. Note that the horizontal and vertical scale factors are 1/sub_sampling factors in northing and easting.

The command line to generate an image of the points in the geocoded image in SUNraster format is the following. The geocoded SAR image and the image of the points can be displayed with the DISP program ***dis2ras.***

```
ras_clist clist_MAP 19970924_gec.mli.ras 19970924_gec_clist.mli.ras 0.25
0.25 255 255 0 3
```

```
dis2ras 19970924_gec.mli.ras 19970924_gec_clist.mli.ras
```

Figure H1 shows the offset measurements locations for this example.

The command line to generate an image of the points in the radar geometry in SUNraster format is the following. Since the clist_RDC coordinates are for the SLC but we want to generate an image of the points in the MLI raster image, it is therefore necessary to enter the number of looks for the MLI image to scale the locations. The geocoded SAR image and the image of the points can be displayed with the DISP program ***dis2ras.***

```
ras_clist clist_RDC 19970924_seg.mli.ras 19970924_seg_clist.mli.ras 4 4 255
255 0 1
```

```
dis2ras 19970924_seg_clist.mli.ras 19970924_seg.mli.ras
```

Measuring the offsets requires an offset parameter file to contain any linear a priori offset model. These model coefficients can be estimated in RDC coordinates using results from ***offset_pwr***. In the example, the images are already co-registered, therefore the initial offset is 0 in both range and azimuth. Offsets are then measured at the RDC coordinates stored in the clist_RDC file. Output is written to the offset_clist file in text format and to the offs_clist and snr_clist files as binary floating point numbers.

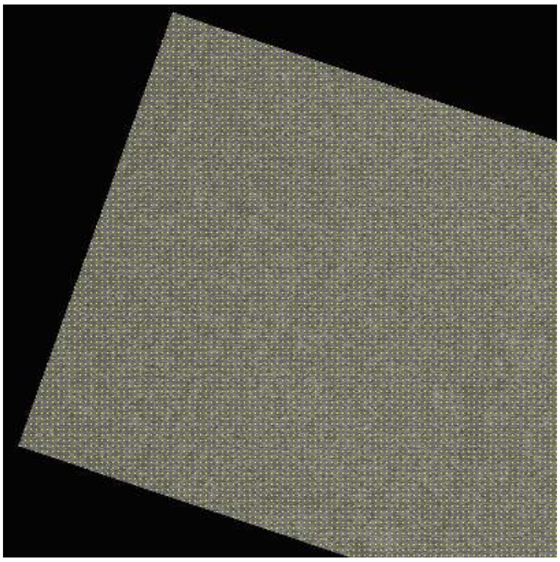


Figure H1. Section of geocoded image 19970924_gec_clist.mli.ras showing offset measurement locations in map coordinates.

To generate the offset parameter file use the program ***create_offset*** as follows accepting all default values

```
create_offset 19970924_seg.slc.par 19971018_seg.slc.par off0 1
```

then compute offsets between SLC images with ***offset_pwr*** as follows

```
offset_pwr    19970924_seg.slc    19971018_seg.slc    19970924_seg.slc.par
19971018_seg.slc.par off0 offs snr 64 64 offsets
```

and finally measure offsets at the specific locations using SNR threshold of 6.5

```
offset_pwr_list   19970924_seg.slc   19971018_seg.slc   19970924_seg.slc.par
19971018_seg.slc.par off0 clist_RDC clist_MAP offs_clist snr_clist 724 315
64 64 offsets_clist 2 6.5
```

Displaying the offset information can be done by conversion of the FCOMPLEX format output raster files to separate range and azimuth offset files and displayed using the DISP program ***rashgt*** with a wrap of 3 pixels.

The conversion of the complex-valued raster files to obtain range and azimuth files is done with the program ***cpx_to_real***. The offsets lists are stored in the files offs_clist_r and offs_clist_az

```
cpx_to_real offs_clist offs_clist_r 724 0

cpx_to_real offs_clist offs_clist_az 724 1
```

Then a multi-look image that is 4 looks in easting + northing is generated to match the s_north and s_east values in dem_RDC_list. This is done with the program ***reallks*** as follows.

```
reallks 19970924_gec.mli 19970924_gec_4_4.mli 2896 4 4
```

Finally the display at 3 pixels offset/color cycle is done as follows using SUNraster files

```
rashgt offs_clist_r 19970924_gec_4_4.mli 724 1 1 0 1 1 3.0 1.2
```

```
rashgt offs_clist_az 19970924_gec_4_4.mli 724 1 1 0 1 1 3.0 1.2

dis2ras offs_clist_r.ras offs_clist_az.ras
```

The offset lists can also be displayed directly with the DISP program ***dishgt***

```
dishgt offs_clist_r 19970924_gec_4_4.mli 724 1 1 0 3.0 1.2

dishgt offs_clist_az 19970924_gec_4_4.mli 724 1 1 0 3.0 1.2
```

The image produced from the offset data in map coordinates is shown below in Figures H2 and H3. Note that the range offsets are measured along the diagonal, and the azimuth offsets are in the direction parallel to the short edge. The geocoded offset measurements have a posting of 160 meters in northing and easting. These offset measurements indicate rapid glacier movement primarily in the azimuth direction. Note the azimuth offset discontinuity associated with the crack at the upper edge on the right side of Figure G2.

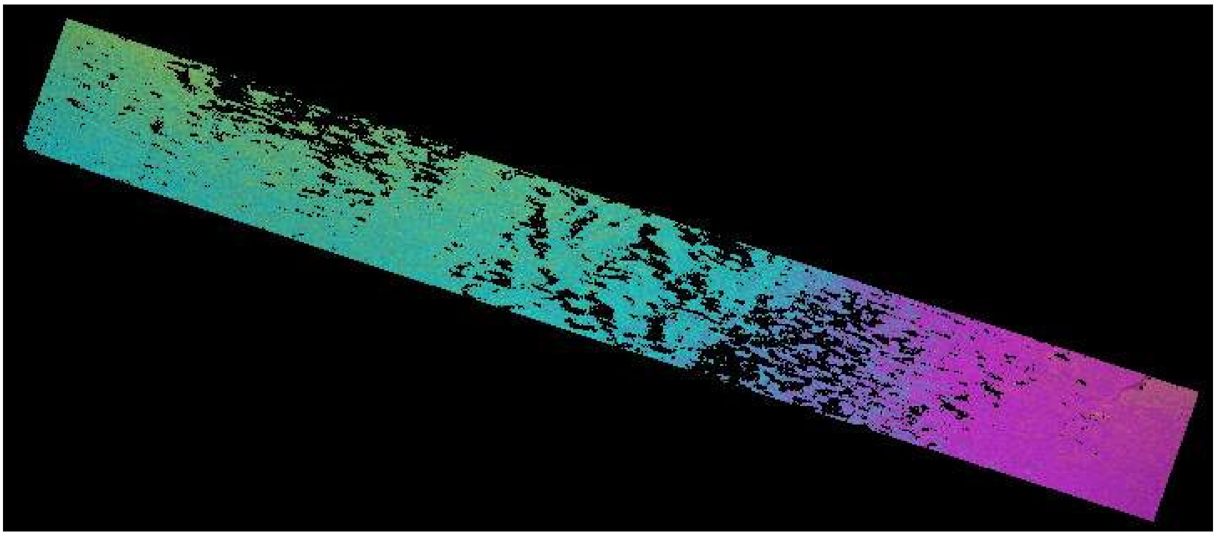


Figure H2. Range offsets offs_clist_r.ras shown at 3 range samples/color cycle

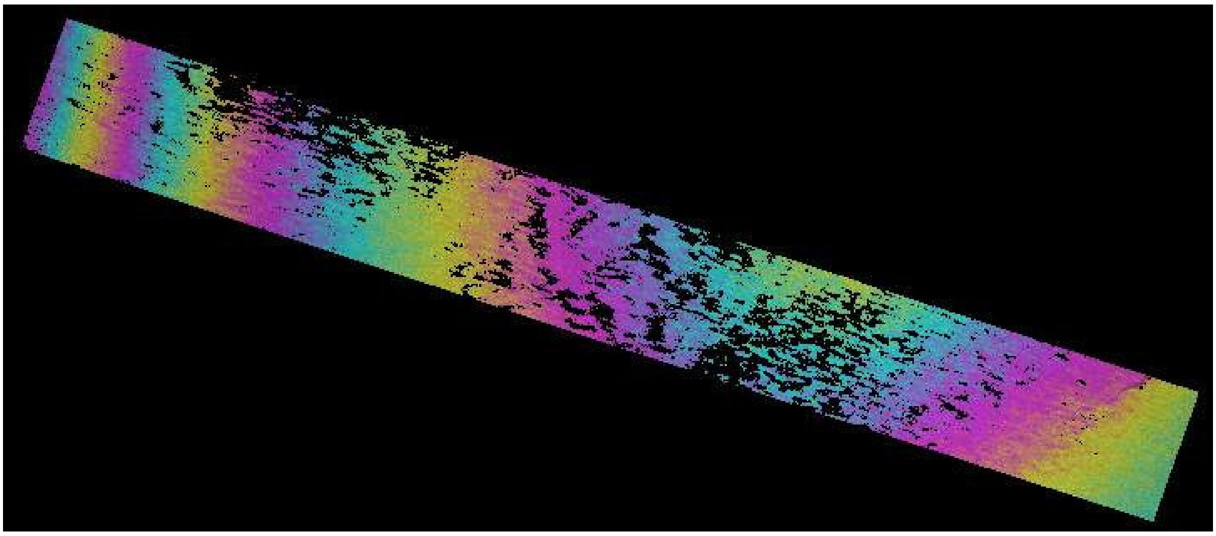


Figure H3. Azimuth offsets offs_clist_az.ras shown at 3 azimuth samples/color cycle

I. Co-registration of two SLCs using a lookup table

In the User's Guide of the ISP module a co-registration method for two SLCs has been presented and discussed. This method was based on cross-correlation of intensities between the two SLC images. In Section 11 an alternative method for co-registering two SLC images has been briefly presented. This method is based on a lookup table that links the geometries of the two images. This method is in principle similar to the one adopted for GTC geocoding. This method requires the availability of a DEM. For more details on the background of the method it is referred to Section 11.

In this Example the processing steps for lookup table-based co-registration are presented in more detail with a practical example based on a pair of SLC images. The results will be compared to a resampled SLC obtained using the traditional method.

For this purpose we consider as reference an ERS-1 image acquired over Las Vegas on 23 May 1996. As file identifier we use the orbit number, 25394. The image is in SLC format. The second SLC was acquired by ERS-2 on 24 May 1996. As file identifier we use the corresponding orbit number, 05721.Aim of this example is co-registering SLC 05721.slc to the reference 25394.slc using both the traditional and the lookup table-based method and compare the results. It can already be said that the two results will be almost identical. On one hand this proves the capability of the co-registration method. On the other hand it must be observed that the dataset is not the best for showing the improved co-registering capability of the look-up table method. This method works better for high resolution datasets, with long baselines and different frequencies (e.g. TerraSAR-X, ALOS PALSAR or ERS-ENVISAT).

For the co-registration, two SLC images and a DEM are required. The table below summarizes the files used in this example. The dataset is available on the DEMO CD-ROM. The SLC image files are in the slc directory. Each image is 2500 pixels wide and 9000 pixels long. The DEM is required for the generation of the lookup table since in this way offsets due to topography can be accounted for. Since the lookup table is generated based on MLI versions of the SLC data, the lookup table must be in the same geometry as the MLI intensity image obtained from the reference SLC image. In this example we use the DEM produced in Example B (Section B.9). This DEM is already in the required format since it has been obtained based on the geometry of the MLI image derived from SLC image 25394.slc. The DEM is available on the DEMO CD-ROM in the dem directory.

Filename	Content
25394.slc	Reference SLC image
25394.slc.par	Reference SLC parameter file
05721.slc	SLC image to be co-registered
25394.slc.par	SLC parameter file of image to be co-registered
25394.dem	DEM in geometry of reference image

Co-registration using the lookup table will consist of the following processing steps

- Generation of MLI intensity images from corresponding SLC files. Generation of the SLC data, including the ISP SLC parameter file and improvement, if possible, must be carried out before generating the MLI image following the indications in the User's Guide of the ISP module.
- Generation of the DEM in SAR geometry. For this it is referred to Example B in this User's Guide.

- Generation of initial co-registration lookup table linking the geometries of the two MLI datasets, considering terrain heights
- Resampling of reference MLI intensity image to the geometry of the second image using the lookup table (forward geocoding)
- Fine registration of the resampled master SAR intensity image with the second SAR intensity image
- Refinement of co-registration lookup table
- Resampling of second SLC to the geometry of the reference SLC using the refined lookup table
- Further refinement to remove residual offsets

The set of command presented in this example can be copied and pasted on the command line to run interactively the co-registration sequence.

I.1. Preparation of files

To allow a comparison between co-registered images obtained with the two approaches, the processing using one or the other approach will be carried out in two separate directories. We will call hereafter the directories as follows:

Co-registration approach	**Directory**
Based on cross-correlation of intensities (traditional)	cross_cor
Based on lookup table	lookup_table

If not available each can be generated with the command mkdir

```
mkdir cross_cor lookup_table
```

For simplicity it is recommended to copy the files needed for processing to each directory.

I.2. Generation of MLI intensity images

Co-registration based on the lookup table requires the availability an image pair in MLI format. If these images are already available, for example for the reference image when generating the DEM in SAR geometry, this part can be skipped.

The MLI images are obtained from the corresponding SLC images using the command ***multi_look*** with multi-look factors 1 in range and 5 in azimuth as follows:

```
multi_look 25394.slc 25394.slc.par 25394.mli 25394.mli.par 1 5
```

```
multi_look 05721.slc 05721.slc.par 05721.mli 05721.mli.par 1 5
```

Each can be displayed with the DISP program ***dispwr*** or they can be displaced simultaneously with the program ***dis2pwr***. With this command we can appreciate the large offsets between the two images:

```
dis2pwr 25394.mli 05721.mli 2500 2500
```

I.3. Generation of DEM in radar geometry

To obtain a DEM in radar geometry it is referred to the procedure presented in Example B. The last output of that example, the DEM in radar geometry in file 25394.dem, has been made available on the DEMO-CD and for this reason the entire processing sequence will be skipped in this example.

I.4. Generation of initial co-registration lookup table

To generate the look-up table linking the geometry of the reference and the second SLC images the program ***rdc_trans*** is used as follows:

```
rdc_trans 25394.mli.par 25394.dem 05721.mli.par lt0
```

From the ISP SLC parameter files of the two SLC images and the DEM in radar geometry the co-registration lookup table lt0 is generated. The lookup table contains the range and azimuth pixel numbers of the second MLI with respect to the associated pixel in the reference MLI. Offsets due to topographic relief are accounted for by using terrain height in the derivation of the lookup table. This is not the case with the traditional offset computation method. It is therefore clear that the lookup table approach is most powerful in case of long baselines and high resolution data.

I.5. Resampling of reference MLI to second image geometry

The initial lookup table might contain errors due to inaccuracy of the orbital state vectors and of the DEM. For this reason a refinement of the lookup table is required. In a similar way to the procedure presented for GTC geocoding (see Example B), the refinement is based on the resampling of the reference MLI image to the geometry of the other MLI image, then offsets between the two images are estimated and these are used to improve the lookup table.

Resampling of the reference MLI image to the geometry of the second MLI image is done with the program ***geocode*** as follows

```
geocode lt0 25394.mli 2500 mli0 2500 1800 2 0
```

The lookup table lt0 is used to resample the reference MLI image 25394.mli to the geometry of the MLI image 05721.mli. The resampled MLI is called simply mli0. It is an intermediate product without relevance for the final outcome of the co-registration procedure. The reference MLI is resampled to the same size of the second MLI image, i.e. 2500 pixel wide and 1800 pixels long. To retain as much information as possible an interpolation based on the square root of (1/distance) is used.

I.6. Offset computation between resampled reference and second MLI

To obtain the field of offsets between the reference MLI resampled to the geometry of the second MLI image and this image, we use the procedure based on the sequence ***create_diff_par***, ***init_offsetm***, ***offset_pwrm***, ***offset_fitm***.

At first we create a DIFF/GEO parameter file called diff0 as follows

```
create_diff_par 05721.mli.par - diff0 1
```

Then an initial estimate of the offsets in range and azimuth is computed. This is stored in the diff0 file.

```
init_offsetm mli0 05721.mli diff0 1 1
```

The program estimates an initial offset of 0 pixels in range and 4 in azimuth showing that there are inaccuracies in the input data and need to be taken care of.

The initial estimate is used to drive the estimation of the field of offsets all over the image. In this example we accept default values for the window size, the number of windows and the threshold. The oversampling factor is set to 2. The field of offsets is save to the file offs0 (binary version) and offsets0 (text version). The binary file snr0 contains the field of SNR values for the corresponding field of offsets.

```
offset_pwrm mli0 05721.mli diff0 offs0 snr0 256 256 offsets 2 16 16
```

A look at the text file offsets0 reveals a large and homogeneous set of offsets all across the image. The field of offsets between the MLIs is then used to estimate the coefficients of the range and azimuth polynomials. Here we use 4 coefficients for each polynomial.

```
offset_fitm offs0 snr0 diff0 coffs0 coffsets0 7. 4
```

The offset statistics show a very good result in azimuth (std. deviation ~ 0.01) whereas for the range direction the std. deviation is larger being ~ 0.09. An additional round of ***offset_pwrm/offset_fitm*** can be used with more windows of smaller size. In this example however we proceed with the refinement of the look-up table. Mismatches will be dealt with later on in a further refinement step based this time on the SLC data.

I.7. Refinement of co-registration lookup table

To refine the co-registration lookup table we use the program ***gc_map_fine***. The program will generate the lookup table lt1 based on the polynomial model stored in the DIFF/GEO parameter file diff0.

```
gc_map_fine lt0 2500 diff0 lt1
```

I.8. Resampling of second SLC to reference SLC

Using the refined lookup table we can now resample the second SLC. This is done with the program ***SLC_interp_lt*** as follows

```
SLC_interp_lt 05721.slc 25394.slc.par 05721.slc.par lt1 25394.mli.par
05721.mli.par - 05721.rslc0 05721.rslc0.par
```

The resampled SLC is called *.rslc0 to indicate that a further refinement will be considered. Based on the lookup table and on values at a number of locations, the program computes an offset polynomial for the range and the azimuth direction respectively, which are then applied to resample the SLC. The program provides an indication on the accuracy of the resampling in terms of standard deviation values of the offset model in a least squares sense. In this example the offset model fit standard deviations is 0.0069 (pixels) in range and 0.0222 (pixels) in range.

I.9. Further refinement of resampled SLC

Although the resampled SLC could be considered acceptable in most cases, it is recommended to run through an offset estimation process and see whether residual offsets are present and eventually can be corrected for.

The refinement consists in

- Generation of an offset parameter file
- Estimation of a field of residual offsets in range and azimuth directions
- Estimation of a range and an azimuth polynomial that model the residual offsets
- New resampling of the original SLC using the lookup table AND the model of residual offsets

Contrarily to the procedure described in Sections I.6 where MLIs were used to estimate offsets, here we will make use of data in SLC format, which means that the programs used will be those of the ISP module.

The ISP offset parameter file that will contain information on the residual offsets is generated with the program ***create_offset*** as follows

```
create_offset 25394.slc.par 05721.rslc0.par off 1
```

Since we are initializing the file, we accept the default values

The field of residual offsets is obtained with ***offset_pwr*** as follows

```
offset_pwr 25394.slc 05721.rslc0 25394.slc.par 05721.rslc0.par off offs snr
128 128 offsets 2 32 64
```

The field of offsets is stored in the binary file called off, whereas the list of SNR values is stored in the file snr. The corresponding text version is called offsets.

The offset model polynomials are obtained with ***offset_fit*** as follows

```
offset_fit offs snr off coffs coffsets 7.0 4
```

The file off contains now the residual offset polynomials, which are non-zero. The accuracy report shows a very good fit with standard deviation of the model fit equal to ~ 0.02 pixels in range and ~0.04 in azimuth (based on more than 1700 samples out of 2048).

With the polynomials of the residual offsets and the refined lookup table it is now possible to obtain an improved version of the resampled SLC. This is done with ***SLC_interp_lt*** as follows

```
SLC_interp_lt 05721.slc 25394.slc.par 05721.slc.par lt1 25394.mli.par
05721.mli.par off 05721.rslc 05721.rslc.par
```

The resampled SLC 05721.rslc can now be compared to the original SLC with the DISP program ***dis2SLC***.

```
dis2SLC 25394.slc 05721.rslc 2500 2500
```

I.10. Co-registration with traditional cross-correlation algorithm

The procedure for co-registering two SLCs using the cross-correlation algorithm has been thoroughly described in the ISP User's Guide. For this reason, we report here only the commands used for this example

If running this example command by command, it is reminded to move to the directory dedicated to the co-registration with the cross-correlation algorithm where we have the original SLC data.

Generation of the ISP offset parameter file with the program ***create_offset***

```
create_offset 25394.slc.par 05721.slc.par off 1
```

Estimation of initial offsets with the program ***init_offset***

```
init_offset 25394.slc 05721.slc 25394.slc.par 05721.slc.par off 1 5
```

Estimation of the the field of offsets with the program ***offset_pwr***

```
offset_pwr 25394.slc 05721.slc 25394.slc.par 05721.slc.par off offs snr 256
256 offsets 2 8 16
```

Generation of offset polynomial with the program ***offset_fit***

```
offset_fit offs snr off coffs coffsets 7 3
```

Further estimation of offsets with ***offset_pwr***

```
offset_pwr 25394.slc 05721.slc 25394.slc.par 05721.slc.par off offs snr 128
128 offsets 2 16 32
```

Refinement of the offset polynomial with ***offset_fit***

```
offset_fit offs snr off coffs coffsets 7 4
```

The offset error statistics show results that are very similar to what obtained with the lookup table approach.

Resampling of the second SLC to the geometry of the reference SLC with the program ***SLC_interp***

```
SLC_interp 05721.slc 25394.slc.par 05721.slc.par off 05721.rslc
05721.rslc.par
```

The resampled SLCs using the two approaches can now be compared with the program dis2SLC. For this it is necessary to move to the parent directory first.

```
cd ..
```

```
dis2SLC lookup_table/05721.rslc cross_cor/05721.rslc 2500 2500
```

We can see that the two SLCs match perfectly, showing that the two procedure perform equally well. It has to be reminded that this was not the best dataset for showing the better performance of the lookup table approach compared to the cross-correlation algorithm.

Figure I1 shows the impact of topography on the offset estimation. The image shows the offset field in range for a PALSAR image pair acquired with 46 day difference and a perpendicular baseline of approximately 2 km over the Austrian Alps. The offsets field in range is color code and overlaid onto the SAR intensity image. The Figure clearly shows the dependence of the range offsets upon topography. Here one color cycle corresponds to one pixel offset. The largest offsets are found in steep relief areas. Inclusion of topographic information, as implemented in the lookup table approach, has the effect of limiting topography effects on the offset estimation.

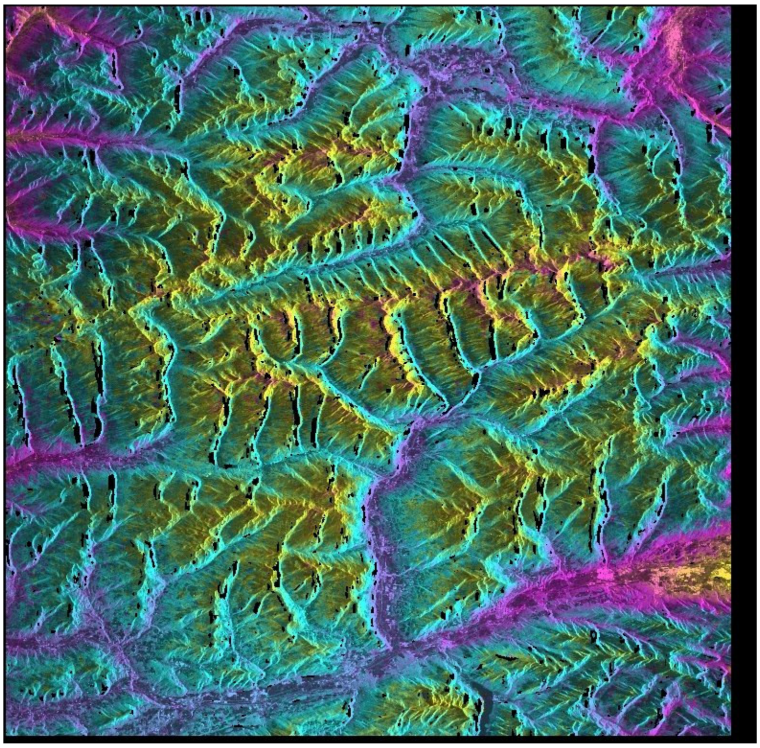


Figure I1. Color coded field of offsets in range for an ALOS PALSAR image pair with almost 2 km perpendicular baseline acquired over the Austrian Alps. The image background corresponds to the SAR intensity image.