Alaska Satellite Facility
software tools
Manual

by
Rüdiger Gens and Tom Logan
TABLE OF CONTENTS

TABLE OF CONTENTS........................................................................................................ III

LIST OF ACRONYMS AND ABBREVIATIONS ........................................................... VI

LIST OF FILE EXTENSIONS..................................................................................... VIII

1. HOW TO USE THIS MANUAL?.................................................................................. 1

2. BRIEF INTRODUCTION TO SAR INTERFEROMETRY........................................ 3
   2.1 HISTORY .............................................................................................................. 3
   2.2 RELEVANT RADAR AND INTERFEROMETRY PARAMETERS............................. 4
   2.3 TECHNIQUES ...................................................................................................... 8
   2.4 INTERFEROMETRIC PRODUCTS ........................................................................ 9
   2.5 APPLICATIONS ................................................................................................ 10
   2.6 REFERENCES ......................................................................................... 13

3. PROCESSING ............................................................................................................. 17
   3.1 DATA SETS ........................................................................................................ 17
   3.2 SAR PROCESSING............................................................................................ 18
   3.3 COREGISTRATION ........................................................................................ 19
   3.4 INTERFEROMGRAM / COHERENCE IMAGE GENERATION............................... 19
   3.5 PHASE UNWRAPPING ..................................................................................... 21
   3.6 DEM GENERATION ...................................................................................... 22

4. TUTORIAL ................................................................................................................. 25
   4.1 READING DATA SETS FROM TAPE ................................................................. 25
   4.2 PRE-PROCESSING OF DATA INTO ASF INTERNAL FORMAT................................. 26
   4.3 THE ASF INTERFEROMETRIC SAR PROCESSOR (AISP)..................................... 27
   4.4 GETTING INFORMATION ABOUT DATA ............................................................ 28
   4.5 HAVING A LOOK AT DATA .............................................................................. 29
   4.6 RADIOMETRIC CALIBRATION OF SAR IMAGERY .............................................. 29
   4.7 RESAMPLING ............................................................................................... 30
   4.8 QUALITY OF COREGISTRATION .................................................................... 31
   4.9 BASELINE REFINEMENT .............................................................................. 32
   4.10 PHASE FILTERING ....................................................................................... 32
   4.11 QUALITY ASSESSMENT OF DEMS ................................................................. 33
   4.12 GEOCODING .............................................................................................. 34
   4.13 MAP PROJECTIONS .................................................................................... 35
   4.14 IMAGE MOSAICKING .................................................................................. 36

5. TOOLS .................................................................................................................... 39
   accum_offset ......................................................................................................... 40
   aisp ......................................................................................................................... 41
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_sar_mosaic</td>
<td>43</td>
</tr>
<tr>
<td>auto_swath_mosaic</td>
<td>45</td>
</tr>
<tr>
<td>avg_in_dop</td>
<td>47</td>
</tr>
<tr>
<td>c2p</td>
<td>48</td>
</tr>
<tr>
<td>calc_deltas</td>
<td>49</td>
</tr>
<tr>
<td>calibrate</td>
<td>50</td>
</tr>
<tr>
<td>ceos2raw</td>
<td>52</td>
</tr>
<tr>
<td>coh</td>
<td>54</td>
</tr>
<tr>
<td>concat</td>
<td>56</td>
</tr>
<tr>
<td>concat_dem</td>
<td>57</td>
</tr>
<tr>
<td>concatam</td>
<td>58</td>
</tr>
<tr>
<td>convert2byte</td>
<td>60</td>
</tr>
<tr>
<td>create_dem</td>
<td>62</td>
</tr>
<tr>
<td>create_dem_grid</td>
<td>63</td>
</tr>
<tr>
<td>ddr2meta</td>
<td>64</td>
</tr>
<tr>
<td>dem2phase</td>
<td>65</td>
</tr>
<tr>
<td>dem2seeds</td>
<td>67</td>
</tr>
<tr>
<td>deramp</td>
<td>69</td>
</tr>
<tr>
<td>deskew</td>
<td>70</td>
</tr>
<tr>
<td>deskew_dem</td>
<td>71</td>
</tr>
<tr>
<td>diff_las</td>
<td>72</td>
</tr>
<tr>
<td>dspddr</td>
<td>74</td>
</tr>
<tr>
<td>dump_multi_volume</td>
<td>76</td>
</tr>
<tr>
<td>elev</td>
<td>77</td>
</tr>
<tr>
<td>eleverr</td>
<td>78</td>
</tr>
<tr>
<td>escher</td>
<td>80</td>
</tr>
<tr>
<td>ffitMatch</td>
<td>82</td>
</tr>
<tr>
<td>fico</td>
<td>83</td>
</tr>
<tr>
<td>fit_line</td>
<td>84</td>
</tr>
<tr>
<td>fit_plane</td>
<td>85</td>
</tr>
<tr>
<td>fit_quadratic</td>
<td>86</td>
</tr>
<tr>
<td>fit_warp</td>
<td>87</td>
</tr>
<tr>
<td>geocode</td>
<td>88</td>
</tr>
<tr>
<td>igram</td>
<td>90</td>
</tr>
<tr>
<td>las_op</td>
<td>91</td>
</tr>
<tr>
<td>las2geotiff</td>
<td>93</td>
</tr>
<tr>
<td>las2jpeg</td>
<td>94</td>
</tr>
<tr>
<td>las2ppm</td>
<td>95</td>
</tr>
<tr>
<td>lz2raw_flywheel</td>
<td>96</td>
</tr>
<tr>
<td>makeddr</td>
<td>98</td>
</tr>
<tr>
<td>meta2ddr</td>
<td>100</td>
</tr>
<tr>
<td>metadata</td>
<td>101</td>
</tr>
<tr>
<td>ml</td>
<td>104</td>
</tr>
<tr>
<td>p2c</td>
<td>106</td>
</tr>
<tr>
<td>phase_filter</td>
<td>107</td>
</tr>
<tr>
<td>projprm</td>
<td>108</td>
</tr>
</tbody>
</table>
# LIST OF ACRONYMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AISP</td>
<td>ASF Interferometric SAR processor</td>
</tr>
<tr>
<td>ASF</td>
<td>Alaska Satellite Facility</td>
</tr>
<tr>
<td>AVHRR</td>
<td>Advanced Very High-Resolution Radiometry</td>
</tr>
<tr>
<td>AVS</td>
<td>Advanced Visualization System</td>
</tr>
<tr>
<td>CCRS</td>
<td>Canadian Centre for Remote Sensing</td>
</tr>
<tr>
<td>CCSD</td>
<td>Computer Compatible Signal Data</td>
</tr>
<tr>
<td>CDF</td>
<td>Common Data Format</td>
</tr>
<tr>
<td>CEOS</td>
<td>Committee on Earth Observation Satellites</td>
</tr>
<tr>
<td>CSA</td>
<td>Canadian Space Agency</td>
</tr>
<tr>
<td>DAAC</td>
<td>Distributed Active Archive Center</td>
</tr>
<tr>
<td>DADS</td>
<td>Data Archive and Distribution System</td>
</tr>
<tr>
<td>DEM</td>
<td>Digital Elevation Model</td>
</tr>
<tr>
<td>DIS</td>
<td>Data and Information System</td>
</tr>
<tr>
<td>DCS</td>
<td>Direct Capture System</td>
</tr>
<tr>
<td>ECS</td>
<td>EOSDIS Core System</td>
</tr>
<tr>
<td>EDC</td>
<td>EROS Data Center</td>
</tr>
<tr>
<td>EDHS</td>
<td>ECS Data Handling System</td>
</tr>
<tr>
<td>EOS</td>
<td>Earth Observing System</td>
</tr>
<tr>
<td>EOSDIS</td>
<td>Earth Observing System Data and Information System</td>
</tr>
<tr>
<td>EROS</td>
<td>Earth Resources Observation System</td>
</tr>
<tr>
<td>ERS-1</td>
<td>First European Remote Sensing Satellite</td>
</tr>
<tr>
<td>ERS-2</td>
<td>Second European Remote Sensing Satellite</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>GCDIS</td>
<td>Global Change Data and Information System</td>
</tr>
<tr>
<td>GCMD</td>
<td>Global Change Master Directory</td>
</tr>
<tr>
<td>GI</td>
<td>Geophysical Institute</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSFC</td>
<td>Goddard Space Flight Center</td>
</tr>
<tr>
<td>HLS</td>
<td>Hue-Lightness-Saturation color model</td>
</tr>
<tr>
<td>HRPT</td>
<td>High Resolution Picture Transmission</td>
</tr>
<tr>
<td>IIAS</td>
<td>Interactive Image Analysis System</td>
</tr>
<tr>
<td>IMS</td>
<td>Information Management System</td>
</tr>
<tr>
<td>INS</td>
<td>Inertial Navigation System</td>
</tr>
<tr>
<td>InSAR</td>
<td>Interferometric SAR</td>
</tr>
<tr>
<td>JERS-1</td>
<td>First Japanese Earth Resources Satellite</td>
</tr>
<tr>
<td>JPL</td>
<td>Jet Propulsion Laboratory</td>
</tr>
<tr>
<td>L0</td>
<td>Raw SAR data (level zero)</td>
</tr>
<tr>
<td>L1A</td>
<td>Processed SAR data</td>
</tr>
<tr>
<td>L1B</td>
<td>Processed and geocoded SAR data</td>
</tr>
<tr>
<td>LaRC</td>
<td>Langley Research Center</td>
</tr>
<tr>
<td>LAS</td>
<td>Land Analysis System</td>
</tr>
<tr>
<td>LZP</td>
<td>Level Zero Processor</td>
</tr>
<tr>
<td>MOA</td>
<td>Memorandum of Agreement</td>
</tr>
<tr>
<td>Acronym</td>
<td>Abbreviation</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MOS-1</td>
<td>Marine Observation Satellite (Japan)</td>
</tr>
<tr>
<td>MOU</td>
<td>Memorandum of Understanding</td>
</tr>
<tr>
<td>MSFC</td>
<td>Marshall Space Flight Center</td>
</tr>
<tr>
<td>MSS</td>
<td>Multispectral Scanner (Landsat)</td>
</tr>
<tr>
<td>MTPE</td>
<td>Mission to Planet Earth</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>NASDA</td>
<td>National Space Development Agency (Japan)</td>
</tr>
<tr>
<td>NCAR</td>
<td>National Center for Atmospheric Research</td>
</tr>
<tr>
<td>NOAA</td>
<td>National Oceanic and Atmospheric Administration</td>
</tr>
<tr>
<td>NRL</td>
<td>Naval Research Laboratory</td>
</tr>
<tr>
<td>NSF</td>
<td>National Science Foundation</td>
</tr>
<tr>
<td>NSIDC</td>
<td>National Snow and Ice Data Center</td>
</tr>
<tr>
<td>RADARSAT</td>
<td>Canadian Synthetic Aperture Radar Satellite</td>
</tr>
<tr>
<td>RAR</td>
<td>Real Aperture Radar</td>
</tr>
<tr>
<td>SAR</td>
<td>Synthetic Aperture Radar</td>
</tr>
<tr>
<td>SeaWiFS</td>
<td>Sea-viewing Wide Field-of-view Sensor</td>
</tr>
<tr>
<td>SIR-B</td>
<td>Shuttle Imaging Radar (B-Mission)</td>
</tr>
<tr>
<td>SIR-C</td>
<td>Shuttle Imaging Radar (C-Mission)</td>
</tr>
<tr>
<td>SLC</td>
<td>Single Look Complex</td>
</tr>
<tr>
<td>SPAN</td>
<td>Space Physics Analysis Network</td>
</tr>
<tr>
<td>SPOT</td>
<td>Systeme Pour l’Observation de la Terre</td>
</tr>
<tr>
<td>STF</td>
<td>SKY Telemetry Format</td>
</tr>
<tr>
<td>TCE</td>
<td>Time Correlation Element</td>
</tr>
<tr>
<td>UAF</td>
<td>University of Alaska Fairbanks</td>
</tr>
<tr>
<td>UCAR</td>
<td>University Corporation for Atmospheric Research</td>
</tr>
<tr>
<td>USGS</td>
<td>United States Geological Survey</td>
</tr>
<tr>
<td>UTM</td>
<td>Universal Transverse Mercator</td>
</tr>
</tbody>
</table>
LIST OF FILE EXTENSIONS

.000 – complex binary level zero swath data
.amp – amplitude image
.chop – chop file
.cpx – complex binary data in ASF internal format
.D – CCSD data file
.dat – old CCSD data file
.ddr – data descriptor file
.dem – digital elevation model file
.fmt – data format file
.ht – elevation file
.img – image data file
.in – AISP processing parameter file
.L – CCSD leader file
.ldr – CEOS leader file
.log – level zero STF processing log file
.meta – metadata file
.nul – CEOS null file
.par – level zero parameter file
.phase – phase image
.pi – processing information file
.proj – map projection parameter file
.raw – CEOS level zero frame data file / ASF internal raw signal data file
.replica – reference function file (Radarsat)
.ti – level zero processing information file
.trl – CEOS trailer file
.vol – CEOS volume descriptor file
1. HOW TO USE THIS MANUAL?

This chapter provides information on how to make use of this manual in an efficient way.

A brief introduction to SAR interferometry provides the user with the knowledge required for SAR interferometric processing. A basic understanding of the imaging geometry and the problems involved helps in following the complex processing flow. A variety of applications is briefly discussed to show the large potential of SAR interferometric data products for the geoscientific community. The given references allow for further study in order to gain a deeper knowledge in the field of SAR interferometry.

The processing chapter gives an introduction to the interferometric processing. It describes the processing flow and provides further information on how the algorithms are actually implemented.

The tutorial gives some hands on experience with the tools. The processing chain is split into small parts that can be managed in a short time. Each processing step is explained in detail.

Further details of all ASF software tools, including those involved in the SAR interferometric processing, are given in a separate chapter. Program descriptions and usages are given as well as sample outputs. The alphabetic order of the commands simplifies the search process.

A glossary explains some reference terms used in the field of SAR interferometry and provides further technical information about the software.

In the appendices, an introductions to the visualization tools SARview and XID as well as further information about software standards and file formats are provided.

The way this manual is written, it serves different purposes:
1. It provides the theoretical background of SAR interferometry and can be used as a reference for that.
2. It explains the processing steps involved in producing a digital elevation model from two SAR images.
3. It serves as a reference book to find information on the processing and usage of a specific tool.

The terrain correction tools are a standalone set of tools, therefore not included in this manual. The create_dem driver program is only described with its general usage. For more information about the processing chain, the configuration file and the functionality consult the create_dem manual.
2. BRIEF INTRODUCTION TO SAR INTERFEROMETRY

This introduction provides the basic understanding of SAR interferometry required to follow the interferometric processing flow to produce a digital elevation model. Large parts of this introduction have been extracted from Gens (1998).

2.1 HISTORY

Imaging radar systems can be separated into two different categories: real aperture radars and synthetic aperture radars. The real aperture radar (RAR), also referred to as brute force system or non-coherent radar, requires a long antenna and a high power output in order to achieve an acceptable resolution and dynamic range since the resolution is proportional to the length of the antenna but inversely proportional to the range. The synthetic aperture radar (SAR), also known as coherent radar, overcomes the limitation of the antenna size by synthesizing an antenna which receives a series of reflected waves and electronically combines them with reference wavelengths. The resolution of a SAR effectively remains the same over all ranges.

SAR interferometric data can provide information on three-dimensional objects. These data are derived from the phase content of the complex radar signal. The use of SAR interferometry can be traced back to the 1960s when the American military used an airborne system for mapping the Darien province in Panama at a scale of 1:250000 by means of radar interferometry. The first published results using this information source for the observation of the surface of Venus and the Moon are given by Rogers and Ingalls, 1969.

Graham (1974) introduced this synthetic aperture radar technique for topographic mapping. He showed that SAR interferometry, with side-looking airborne or spaceborne geometry, can be used to create topographic maps for two reasons. The resolution of some SAR data is adequate to identify various features and objects of the terrain. In addition to this, a sufficient number of points can be measured using the interferometric geometry to describe a terrain surface in detail.

After this theoretical study it was not until 1986 before the first practical studies, carried out at the Jet Propulsion Laboratory (JPL) in Pasadena, California, were published by Zebker and Goldstein. Two SAR antennas were mounted on an aircraft. One antenna transmitted a radar signal and the backscattered signal was received by both antennas simultaneously. A correction for the aircraft roll was not applied due to the lack of suitable data on the aircraft attitude. Therefore, the achieved accuracy was still quite limited. Nevertheless, it showed what the limitations of the interferometric technique are and how the performance could be improved.

Gabriel and Goldstein (1988) extended the interferometric technique by adapting the existing technique to crossed orbits using interferometric repeat-pass data from the SIR-B
mission inclined by an angle of 1.2 degrees. The angle between the orbits leads to a more complex processing. Besides a Doppler refocusing for the SAR azimuth correlation, the two images could not be overlaid without resampling in range direction. In addition to that, the crossed orbits caused small altitude variations appearing as small shifts in the azimuth direction.

The technique of differential interferometry was introduced by Gabriel et al. (1989). With two interferograms derived from three different SEASAT observations a double-difference interferogram was calculated. The change shown in the differential interferogram was due to swelling of water-absorbing clays in the scene. It was assumed that differential interferometry can detect small elevation changes in the order of 1 cm or even less (Gabriel et al., 1989). That meant that the technique could provide accurate measurements of geophysical phenomena, residual from seismic events, motions from prevolcanic swelling and other such events.

A multibaseline study by Li and Goldstein (1990) presented an error model for topographic mapping. They demonstrated that interferometric data regularly acquired by a spaceborne SAR can provide extremely useful topographic information. This study also showed that the sensitivity of the height measurements increases with the length of the baseline. At the same time the phase error also increases.

Since the launch of ERS-1 in July 1991, a large number of interferometric data sets acquired in C-band have become available. The ERS-1 satellite was the first operational spaceborne system which acquired radar data suitable for SAR interferometry on a routine basis. With these data sets it was possible to investigate the potentials and limitations of SAR interferometry. With the availability of data sets from ERS-2 launched in April 1995, the European Space Agency (ESA) could set up the tandem mission which combined data sets from ERS-1 and ERS-2 acquired only one day apart. The tandem data sets enabled more detailed investigations on the optimal performance of spaceborne systems in terms of temporal decorrelation and atmospheric effects. All these efforts led to an increasing number of publications in various fields.

### 2.2 Relevant Radar and Interferometry Parameters

A synthetic aperture radar is an active sensor transmitting and receiving microwave signals, i.e. measuring distances between the sensor and the point on the Earth’s surface, where the signal is backscattered, perpendicular to the flight direction. This distance is defined as slant range which can be projected on the ground representing the ground range.

The flight direction is also referred to as along-track or azimuth direction, whereas the direction perpendicular to the flight path is defined as across-track or range direction. The angle between the direction the antenna is pointing and the nadir is called look angle. The angle between the radar beam center and the normal to the local topography is referred to as incidence angle. Both angles are sometimes used synonymously which is only valid if the InSAR geometry is simplified neglecting the Earth’s curvature and the
local topography. The look angle of the sensor is one of the main parameters determining the viewing geometry and the incidence angle of the backscattered signal. It significantly affects the backscatter behavior of the terrain. Depending on the characteristics of the illuminated terrain, areas of layover and shadow occur in the imagery.

The *wavelength* of the sensor determines the penetration depth of the transmitted signal into the vegetation layer of the terrain surface. The longer the wavelength, the deeper the penetration layer. The energy of an X-band sensor is mainly reflected at the top layer of the canopies whereas most of the L-band signal penetrates through the upper vegetation layer and is reflected at the ground surface. The backscatter behavior of C-band is less predictable. Due to volume scattering effects, the layer of backscattering is less determined and does not correspond directly to a terrain surface, neither the vegetation surface nor the ground surface.

The *spatial resolution* of the radar sensor defines the minimum separation between the measurements the sensor is able to discriminate and determines the amount of speckle introduced into the system. Speckle is a scattering phenomenon which arises because the spatial resolution of the sensor is not sufficient to resolve individual scatterers. The higher the spatial resolution of the sensor the more objects on the ground can be discriminated. The term ‘spatial resolution’ is often confused with the pixel size, which is actually the spacing of the pixels in the azimuth and ground range direction.

The *bandwidth* with which a signal is processed has to be carefully chosen since it controls the phase aliasing and the amount of ambiguities introduced into the signal. In other words, the bandwidth determines the focusing of the signal.

As mentioned above, SAR interferometry is a technique for extracting information about three-dimensional objects from complex radar signals. The real (Re) and imaginary (Im) part of the complex values contain information about the *amplitude* \( a \), as well as the *phase* \( \varphi \). This information can be extracted from the complex values by

\[
\varphi = \arctan \frac{\text{Im}}{\text{Re}},
\]

\[
a = \sqrt{\text{Im}^2 + \text{Re}^2}.
\]

The phase information of two complex radar images is combined using SAR interferometric techniques. The phase difference of the two images is calculated for an interferogram where *fringes* represent the whole range of the phase from 0 to \( 2\pi \) in a full color cycle. The correlation of the phase information of two corresponding pixels is measured as *coherence* in the range from 0 to 1. The phase coherence can be decorrelated due to thermal noise introduced by the system, changes in the geometry affecting the baseline or temporal changes between the data acquisition.

The general InSAR geometry, simplified by neglecting the Earth’s curvature, is illustrated in Figure 1.

Two antennas \( O_1 \) and \( O_2 \) on ideally parallel flight paths are separated by a baseline \( B \). From both antennas the slant ranges \( r_1 \) and \( r_2 \) to the same surface element are measured. With the look angle \( \theta \) and the flying height \( H \) the geometry is fixed.
Figure 1: General geometry of SAR interferometry (Gens, 1998)

The height of the point \( z(x,y) \) can be determined by

\[
z(x, y) = H - r_1 (\cos \xi \sqrt{1 - \sin^2 (\theta - \xi)} - \sin \xi \sin(\theta - \xi)) \\
= H - r_1 (\cos \xi \cos(\theta - \xi) - \sin \xi \sin(\theta - \xi)) \\
= H - r_1 \cos(\xi + \theta - \xi) = H - r_1 \cos \theta \quad (2-2)
\]

Instead of the look angle, the baseline tilt angle \( \xi \) can be used for the calculation. In that case, the term \( \sin (\theta - \xi) \) can be derived from the interferometric range difference and the baseline.

Apart from the intensity value indicating how much backscattered energy is received by the radar antenna, the data sets used for the InSAR technique also provide phase information which is an ambiguous representation of the range. The phase difference \( \phi \) between the two radar signals received from the same surface element at the two antenna positions is

\[
\varphi = \frac{4\pi}{\lambda} (r_2 - r_1) = \frac{4\pi}{\lambda} (B_y \sin \theta - B_z \cos \theta). \quad (2-3)
\]

Here the wavelength \( \lambda \) and the ranges \( r_1 \) and \( r_2 \) can also be described in terms of the baseline components \( b_y \) and \( b_z \) and the look angle \( \theta \). This formula is valid for the satellite
and the airborne configuration, addressed in the following paragraphs, under the assumption that in the airborne approach both antennas transmit and receive their own signals. This approach effectively doubles the physical baseline.

The key element of this interferometric geometry is the baseline. The baseline, defined as the separation between two antenna positions either mounted on an aircraft or realized by two ideally parallel satellite orbits, can be represented in three different ways, as shown in Figure 2.

![Figure 2: Possible baseline representations (Gens, 1998)](image)

The baseline can be described by its length B and the orientation angle \( \alpha \) or by dividing the baseline in two components, either in the horizontal \( (B_y) \) and the vertical \( (B_z) \) component or the components \( (B_x) \) and \( (B_\alpha) \) of the baseline, which are parallel and perpendicular to the range direction, respectively. All three representations can be found in the literature.

According to Small et al. (1996), there are several ways for estimating the InSAR baseline. Within flat areas the normal baseline component can be derived from the local fringe frequency. Furthermore, the baseline can be calculated using an orbit to orbit approach for each point in the reference orbit. This is based on a closest approach criterion or zero tangential component. Small et al. (1993) described an iterative non-linear least squares fit using tie points and the unwrapped phase in order to adjust the baseline model as well as the phase constant. A triangulation of the slant ranges to a point on a reference ellipsoid making use of the range image offset was developed by Pasquali (1995). Baselines for all points on a coarse grid distributed over the scene were calculated by Schwäbisch (1995) using image simulation based on precise orbits and a reference ellipsoid.

The length of the baseline determines the suitability of the data set for a particular application. With an increasing length of the baseline the phase noise leads to a decorrelation and a lower level of coherence. The coherence is lost completely if the baseline reaches its critical length. This critical baseline \( B_c \) calculated as
depends on the wavelength $\lambda$, the range $r$, the resolution in range $R_y$ and the look angle $\theta$. An optimal baseline as such does not exist. The dependency on the above mentioned system parameters leads to a trade off between the level of noise introduced in the data sets and the sensitivity of the phase to height changes.

Ferretti et al. (1996) proposed a multibaseline technique for improving the quality of interferometric results. They showed that the combination of more than two SAR images provides a robust technique for detecting and reducing atmospheric effects. Artifacts occurring in single interferograms are reduced by averaging the uncorrelated atmospheric contributions. This combination also provides an “atmospheric” noise map for each interferogram and an averaged coherence image that gives a measure of the signal-to-noise ratio (SNR) on a fine spatial resolution (Ferretti et al., 1996).

An analysis by Zebker et al. (1994b) about the accuracy of topographic maps derived from ERS-1 data indicated that the baseline length is one of the potential error sources. Precise knowledge of the baseline geometry is needed because an error in the baseline angle cannot be distinguished from a slope on the terrain surface.

2.3 TECHNIQUES

The three main ways of acquiring SAR interferometric data are across-track, along-track and repeat-pass interferometry.

The across-track method requires two SAR antennas mounted on the same platform for simultaneous data acquisition. This technique is only employed on airborne systems, although studies for the implementation by means of a tethered satellite system, in which two vertically spaced physical antennas are connected by a tether and are carried along parallel paths by a deployer and a sub-satellite, have been carried out. The theoretical background is described by Moccia and Vetrella (1992), who also developed a mathematical model for this approach (Moccia et al., 1995). Once the phase unwrapping has successfully been performed, the elevation map can be derived. The main problem with this geometry in the airborne configuration is the fact that errors caused by the aircraft roll cannot be distinguished from the influence of the terrain slope. Since a satellite track is more stable than an airborne flight path, this problem is less critical in the spaceborne case.

At present, the along-track approach is only applicable to airborne SAR systems, as it requires two antennas on the same platform. The geometry of along-track interferometry does not differ significantly from the geometry of across-track interferometry. Only the x- and the y-axes are changed. Therefore, the phase difference $\Phi$ between the corresponding signals is caused by the movement of the measured object, e.g. water currents. The moving surface leads to a Doppler shift according to the phase velocity of the water waves. All stationary targets are not visible whereas the moving ones can be
seen in the radar imagery. For absolute velocity measurements, the phase difference needs to be calibrated due to baseline components in the y- and z-direction caused by aircraft movements (yaw and pitch). This configuration is suitable mainly for the mapping of water currents, the detection of moving objects and the measurement of directional wave spectra (e.g. Orwig and Held, 1992; Goldstein et al., 1989).

The repeat-pass method requires only one antenna that acquires data sets by passing the same area twice, covering it with a slightly different viewing geometry. This approach is based on the precise knowledge of the flight paths and hence is most suited to spaceborne systems.

Differential interferometry, described in more detail by Gabriel et al. (1989), provides relative measures in the order of a few centimeters or even less for movements in the vertical direction, e.g. for change detection. Due to the viewing geometry, differential InSAR is able to measure only the displacement in the range direction. The differential interferogram can be calculated using three images or more. Alternatively, one of the interferograms can be simulated by registering an existing DEM to the geometry of another image pair. In this case the quality of the DEM has a significant impact on the result. For small-scale changes such as land subsidences, it is not possible to use digital elevation models as data source at all. Massonnet et al. (1993), for example, used this approach for the investigation of the Landers earthquake of 1992. Zebker et al. (1994a) carried out a study on the accuracies that can be achieved and the various limitations.

As the phase discontinuities in the two interferograms do not occur at the same position, each interferogram must be unwrapped. The sensitivity of this technique depends on the baseline geometry and can be increased by averaging over pixels at the expense of spatial resolution. With a decreasing baseline length the system noise can be reduced and hence higher accuracies of the detected height differences can be achieved. The wavelength of the sensor influences the resolution of the derived phase information.

### 2.4 Interferometric Products

The interferogram is defined as the product of the complex SAR values of a master image and the complex conjugate of a slave image. The amplitudes of the corresponding pixels are averaged and the difference of the phase values are calculated for each point in the image. The phase difference, given modulo $2\pi$, is color encoded in the fringes.

The determination of the range is achieved by the additional use of the phase information of the complex radar signal. In the interferometric viewing geometry the phase difference is proportional to the range difference. This is the case as long as the signals have a fixed relation in phase, i.e. the signals are coherent. The coherence is a measure for the correlation of the phase information of two corresponding signals and varies in the range of 0 to 1. The degree of coherence can be used as a quality measure because it significantly influences the accuracy of phase differences and height measurements. Several factors such as thermal noise, phase errors due to the processing, slightly different viewing positions, changes in the object phase between the data acquisition, and different atmospheric conditions during the data acquisition can decrease the coherence.
All these factors can even lead to a complete loss of coherence. This also occurs if the baseline exceeds its critical length.

The phase information in the interferogram is directly related to the topography. Unfortunately, this information is given modulo $2\pi$. In order to calculate the elevation of each point, it is necessary to solve this ambiguity, i.e. the correct integer number of phase cycles needs to be added to each phase measurement to obtain the correct slant range distance. This ambiguity solution is referred to as phase unwrapping. The result of this procedure is the **unwrapped phase**.

The unwrapped phase containing the topographic information is converted to a *digital elevation model*.

A *differential interferogram* is generated by the difference of two interferograms and contains information about small-scale displacements that occur between the data acquisitions. The differential interferogram can be produced in two different ways. Based on two phase-unwrapped interferograms, the difference of these interferograms can be calculated. Alternatively, an existing digital elevation model can be registered to the viewing geometry of a calculated interferogram. The result of this approach is a *simulated interferogram*. The difference of the original and simulated interferogram is the required differential interferogram.

As a final step, all relevant interferometric products can be projected to a common reference system to obtain *geocoded products* in order to combine the interferometric results with information from other sources.

### 2.5 Applications

One of the main reasons why the technique of interferometry was adapted to radar imagery, was to use SAR interferometric data for *topographic mapping* (Graham, 1974). By introducing Global Positioning System (GPS) and Inertial Navigation System (INS) measurements, it is possible to determine the aircraft motions more accurately and to generate significantly better topographic maps. Hagberg and Ulander (1993) estimated the optimal baseline for topographic mapping by spaceborne systems. They simulated the altitude error caused by the radar system or by the topography. With the increasing steepness of the terrain slope, phase aliasing leads to layover and shadow effects. The effects of speckle and thermal noise were reduced by averaging before the phase unwrapping was performed. Zebker *et al.* (1994b) estimated the accuracy of topographic maps derived from ERS-1 imagery. The analysis indicated two main errors: the height estimation error, which is a function of the error in phase estimation, and the error in the knowledge of the baseline length. Precise knowledge of the baseline geometry is needed because an error in the baseline angle cannot be distinguished from a slope on the terrain surface (Zebker *et al.*, 1994b).
Several theoretical studies about generating digital elevation models by SAR interferometry have been carried out. Due to the lack of suitable reference DEMs, the quantitative evaluation of the result is often difficult. Contour maps were often used for the comparison with the interferometric data sets (Zebker and Goldstein, 1986; Prati and Rocca, 1990). The main problem of this approach is the identification of ground features in both sources. Cumming and Hawkins (1990) assembled a list of possible error sources in the estimation of terrain elevation. Lanari et al. (1996) generated a digital elevation model derived from multifrequency shuttle mission data. The algorithm combined unwrapped phase patterns from L-, C- and X-band information. The fusion consisted of a bias removal in C- and X-band performed by a Kalman filter, followed by a weighting of the coherence and the wavelength. The final step in this approach was the summation of the three phase information. Ferretti et al. (1997) proposed a multibaseline approach for the automatic generation of high quality DEM. The resolution of the coherence image can be substantially improved and the combination of uncorrelated phase artifacts due to atmospheric effects significantly reduces their impact on the DEM accuracy.

The phase difference in the along-track geometry is caused by the motion of a surface and is used for monitoring, for example, ocean currents. This technique was first implemented by Goldstein and Zebker of JPL in 1987. Measurements from an inertial navigation system are used to correct phase shifts in the data caused by aircraft roll (Goldstein et al., 1989). Another way of estimating the aircraft attitude is the inclusion of a small area of land in the scene (Goldstein and Zebker, 1987). After the correction for aircraft motion the remaining phase difference is assigned to geophysical sources such as wind effects, tidal currents, wave orbital velocities, internal waves and other currents (e.g. Shemer et al., 1993; Thompson and Jensen, 1993). Carande (1994) estimated the coherence time of the ocean surface using a dual-baseline interferometric SAR. Ainsworth et al. (1995) demonstrated that under some conditions even absolute velocities can be determined. Bao et al. (1997) investigated the ocean wave imaging mechanism and derived a velocity bunching model.

In polar research, the InSAR based measurements provide information about flow velocities and tidal displacements as well as grounding line positions. The fringe patterns contain the effects of ice flow motion and tidal action between the times of data acquisition. Various studies were carried out in the Antarctic peninsula (e.g. Goldstein et al., 1993) and in Greenland (e.g. Joughin et al., 1996a; Rignot, 1996). Joughin et al. (1995) stated that InSAR has its potential for the observation of small-scale and regional velocity fields of ice sheets as well as for the detection of changes in ice flow patterns. Kwok and Fahnestock (1996) pointed out that due to the limited number of available data sets for polar regions, only observations of motions along the line of sight of the radar are possible so far. By combining data sets from ascending and descending orbits, an assumption of ice flow direction could be replaced by the more precise estimation using two velocity vectors.

Most of the papers published about the use of SAR interferometric data for monitoring seismic events deal with earthquakes that occurred in California in the first half of the 1990s. Studies on the Landers earthquake of 1992 (Massonnet et al., 1993 and 1994;
Feigl et al., 1995) validated the potential of SAR interferometry for determining coseismic displacements. Results were also reported from the Eureka Valley earthquake in 1993 (Massonnet and Feigl, 1995; Peltzer and Rosen, 1995) and from the Northridge earthquake in 1994 (Massonnet et al., 1996; Murakami et al., 1996). Most of these results agreed well with the conventional measurements. Besides the coseismic displacements, the interferometric data sets provided more insight into the modeling of the earthquake motion, e.g. earthquake rupture processes and fault segmentation (Peltzer et al., 1994) and focal mechanism (Massonnet and Feigl, 1995). The interferograms contain all the coseismic and some of the postseismic deformations. For the investigation of small-scale features in the interferogram, Peltzer et al. (1994) simulated three simple distortions - a rotation about the horizontal and the vertical axis as well as a distributed simple shear. From the orientation of the resulting fringes the direction of the local displacement gradient can be calculated. Zebker et al. (1994a) pointed out one important limitation of the studies previously carried out: the phenomena observed in the interferometric measurements had the same scale as the distortions contained in the reference data such as digital elevation models, GPS measurements etc.

SAR interferometric data have been used in the field of volcanic hazards for monitoring and mapping of lava flows. First results were achieved by observations from TOPSAR airborne data. Evans et al. (1992) assessed the damage caused by pyroclastic flows and lahars, and the intercomparison of volcano morphology. ERS satellite data were successfully used for the monitoring of small surface changes due to volcanic inflation and of magma movements due to volcanic deflation (Massonnet et al., 1995; Wadge et al., 1997). Briole et al. (1997) detected flank deformation due to lava emplacement. InSAR data were used for topographic mapping of volcanoes (Mouginis-Mark and Garbeil, 1993) and the mapping of lava flows (Zebker et al., 1996). Stevens et al. (1997) stated that SAR interferometric data are suitable for the two approaches generally used for mapping lava. Planimetric mapping of the lava volume is calculated by multiplying the lava area by an estimated mean thickness, whereas the more precise topographic mapping of the volume and morphology of the lava is based on a comparison of changes between the pre- and post-emplacement topography.

Various studies have been carried out to derive additional information from InSAR data for areas covered by forestry. Early results based on ERS-1 repeat-pass imagery indicated that the coherence level in scenes covering forested areas was not sufficient for the derivation of additional information. Askne et al. (1997) concluded that the coherence is mainly determined by temporal decorrelation, the volume scattering layer width and the area fill factor. They developed a model to define a relation between the interferometric observations and basic forest properties. Coherence properties can be used to distinguish between forested and non-forested areas. Furthermore, the interferometric effective forest height was estimated by comparing it with an existing digital elevation model (Askne et al., 1997). According to Hagberg et al. (1995), the coherence was found to be sensitive to temperature changes around zero degrees but insensitive to wind speed. Wegmüller and Werner (1995) were even able to distinguish a number of different forest types. Finally, Wegmüller and Werner (1997) studied the derivation of vegetation parameters. This was also extensively investigated by Treuhaft et al. (1996).
The problem of land subsidence due to gas exploration has been the subject of several studies in the Netherlands (van der Kooij et al., 1995). The main problem with these kind of land subsidences is that the amount of change over a year is so small that it is difficult to separate it from atmospheric effects. Compared to the land subsidence in the Netherlands, the changes due to oil exploration in California, investigated by Fielding et al. (1997), were much larger.

2.6 REFERENCES

Overview papers
General theory and applications:


Signal theoretical aspects of InSAR:

Comprehensive review of applications:

Specific papers:


ASF software tools – Brief introduction to SAR interferometry


ASF software tools – Brief introduction of SAR interferometry


3. PROCESSING

In this chapter the SAR interferometric processing flow is described in more detail. After providing some more information on data sets and SAR processing, it goes methodically through each step required for the processing of SAR interferometric data.

3.1 DATA SETS

The data sets used for SAR interferometric processing can be processed to different levels and can be stored in various formats.

After processing the analog SAR signal to binary SAR signal data, the data is called *level zero data* in SKY telemetry format (STF). The level zero data covers a certain area on the ground in the form of a swath. The length of the swath depends on the amount of data originally collected during the actual acquisition. The size of the files varies but can easily reach some gigabytes. The level zero swaths are then subdivided in frames. For ERS imagery, these frames have a size of 100 · 100 km, which is equivalent to about 26000 lines of radar data. The accompanying leader file is defined in CEOS standard format. This is why these data sets are referred to as *CEOS frames*.

In order to work with the swath and frame formats within the ASF software tools, they have to be converted into the *ASF internal raw signal data format* using the programs *lz2raw* and *ceos2raw*, respectively. In this format, all critical parameters are extracted from the parameter or leader file and stored in a processing parameter file (*.in) and metadata file (*.meta).

The following table contains the extensions of the files that should be present with the different data format.

<table>
<thead>
<tr>
<th>Data format</th>
<th>STF</th>
<th>CEOS frames</th>
<th>ASF internal raw</th>
<th>single look complex</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.000</td>
<td>.raw (D)</td>
<td>.raw</td>
<td>.cpx</td>
</tr>
<tr>
<td></td>
<td>.000.chop</td>
<td>.ldr (L)</td>
<td>.in</td>
<td>(.ddr)</td>
</tr>
<tr>
<td></td>
<td>.000.par</td>
<td>.pi</td>
<td>.meta</td>
<td>.in</td>
</tr>
<tr>
<td></td>
<td>.log</td>
<td></td>
<td>.fmt</td>
<td>.meta</td>
</tr>
<tr>
<td></td>
<td>.par</td>
<td></td>
<td>.replica*</td>
<td></td>
</tr>
</tbody>
</table>

* applies only to Radarsat data

**Table 1:** Data format used for SAR interferometric processing
Running the SAR raw signal data through the ASF Interferometric SAR processor (*aisp*) leads to a single look complex (SLC) image, a so-called level one product. Data used for SAR interferometric processing are generally given in complex format. A convenient representation of the complex signal, as it consists of a real and an imaginary part, is a vector whose length represents the amplitude and whose orientation refers to the phase.

There are a couple of tools required to analyze and visualize the data sets at any stage.

Apart from SAR data itself, which is stored in a binary form, the majority of the other files is stored in ASCII format and can be viewed with any editor available on the system. There are a couple of exceptions. The CEOS leader file (.ldr) and the data descriptor file (.ddr) are not completely in ASCII format. In order to display the contents of these files, some tools are provided. The program `metadata` enables the user to read any of the CEOS records and store them in a file for further reference. The older version of the tools created intermediate files in LAS format that had a data descriptor file. The newer version of the tools extracts all the information out of the metadata file. There are two programs (*meta2ddr* and *ddr2meta*) to convert between old and new format. The program `dspddr` displays the contents of the data descriptor file, which stores the required information on the imagery in a LAS compliant format.

All intermediate products are stored in a generic binary format so that they can be viewed with the tools SARview or XID. These programs provide the functionality of a standard visualization tool, which allows a limited analysis of the data sets. They can read LAS or CEOS data. SARview provides the ability to export to JPG format (see Appendices B and C).

### 3.2 SAR PROCESSING

For SAR interferometric purposes, the SAR processor has to meet a couple of requirements. The most obvious is certainly that the phase is preserved during the processing. Otherwise, the signal is not coherent and SAR interferometry is not possible. The other critical parameter of SAR processing is the Doppler centroid used. There are different approaches to selecting the Doppler value used for the SAR processing.

1. Processing the SAR data to zero-Doppler removes the skew in the azimuth direction. Any InSAR image pair processed this way will have the same geometry in the first place.
2. If the two data sets are processed to their individual Doppler values, it is advisable to apply a baseband filtering to the images at a later stage to match the phase spectra.
3. Both images can be processed to an average Doppler value. For SAR interferometry the two images do not necessarily have to be deskewed. They only have to be in the same geometry. It should be pointed out that this method only leads to satisfactory results as long as the Doppler values for both images are not significantly different.
For the SAR interferometric processing of ERS imagery, the average Doppler approach generally yields good results, because the values for the Doppler centroid are usually very close to each other.

The ASF interferometric SAR processor (aisp) allows processing with different Doppler centroid values. At the same time, several other options for applying estimated offsets, processing subsets and creating a number of intermediate results for troubleshooting are provided.

### 3.3 Coregistration

Coregistration is the process of lining up two images, a so-called master image and a slave image, covering the same area in a way that they fit exactly on top of each other.

In order to do that, the offset between the two images needs to be determined. The first estimate is derived from the state vectors (resolve). However, the orbit information is not good enough to determine the offset down to the pixel level. It needs to be refined by a correlation match in the frequency domain (fitMatch) using computationally effective fast Fourier transformations. For the correlation amplitude images, coherence images or complex radar images can be used. Our fine coregistration algorithm (fico) uses the coherence values as default. The approach based on complex radar values is generally more accurate but is, on the other hand, very time consuming.

Once the offset is determined, the slave image is resampled to match the master image. The ASF tools provide three different methods to perform this resampling. The simplest method applies a linear fit along a line for two offsets estimated at different positions (fit_line). This offset is used within the ASF Interferometric SAR processor (aisp). The other techniques determine a translation as well as a rotation for the resampling. One method calculates the transformation coefficients of a linear resampling matrix (fit_plane), whereas the other method creates a horizontal and vertical warp image grid (fit_warp), a method particularly useful for moving or deforming surfaces such as glaciers and ice sheets. The resampling for these two methods is carried out by the program remap. For the interpolation during the resampling a nearest neighbor or a bilinear approach is the simplest to use. Using a sinc function leads to the most accurate results but takes the longest time.

As a rule of thumb, the execution time of the coregistration algorithm increases with the growing complexity of the statistical analysis involved.

### 3.4 Interferogram / Coherence image generation

The interferogram is defined as the product of the complex values of a master image and complex conjugate of a slave image, i.e. the amplitudes of the corresponding pixels are averaged and the difference of the phase values for each point in the image is calculated.
(igram). The phase difference, given modulo $2\pi$, is color encoded in the so-called fringes. A fringe represents the whole range of the phase in the interferogram from $-\pi$ to $\pi$ in a full color cycle, as shown in Figure 3.

The pixel size in the calculated interferogram sometimes needs to be adapted to a square pixel size on the ground. For example, ERS imagery is acquired with a pixel spacing of 20 meters in the range direction and a line spacing of four meters in the azimuth direction. This process, usually referred to as multilooking, reduces the amount of data and noise at the expense of a loss in spatial resolution.

![Interferogram of an area near Fairbanks, Alaska](image)

**Figure 3:** Interferogram of an area near Fairbanks, Alaska

In the interferometric viewing geometry the phase difference is proportional to the range difference. This is the case as long as the signals have a fixed relation in phase, i.e. the signals are coherent. The *coherence* is a measure of the correlation of the amplitude and phase information of two corresponding signals and varies in the range of 0 to 1, as shown in the coherence image (coh) in Figure 4.
Bright parts show a high level of coherence, whereas dark areas indicate a lack of coherence.

**Figure 4:** Coherence image of an area near Fairbanks, Alaska

### 3.5 Phase Unwrapping

In SAR interferometry, the phase is directly related to the surface topography. In order to quantify the surface topography, the wrapped phase values must be unwrapped. We have implemented Goldstein’s branch cut algorithm, the classical path-following method that is computationally very fast and requires little memory. As with all path-following methods, it creates accurate results where the phase ambiguity can be solved and leaves holes where no solution can be found.
In order to improve the performance of the phase unwrapping algorithm (escher), the method has been adapted in the following way. A low-resolution reference DEM, e.g. from the United States Geological Survey (USGS), is used to subtract the terrain induced phase. Furthermore, the phase is filtered (phase_filter) before the phase unwrapping is carried out. After phase unwrapping, the terrain induced phase is added back to the unwrapped phase.

The baseline is refined using the unwrapped phase and seed points generated from the low-resolution reference DEM. This is an iterative process that usually takes a couple of iterations before the solution converges.

The phase unwrapping and baseline refinement are part of the program tandem_ifm that carries out all processing steps required to generate a digital elevation model from an interferogram, further described in the next section.

### 3.6 DEM Generation

In order to generate the digital elevation model, the phase still has to be converted to an elevation (elev) and the slant range geometry has to be transformed into ground range (deskew_dem).

An error map for the slant range DEM can be generated from the coherence image and the unwrapped phase (eleverr).

For further use, it is desirable to geocode the digital elevation model (geocode). This adds coordinates and a map projection to the DEM. A variety of map projections are supported: geographic, polar stereographic, Universal Transverse Mercator (UTM), Lambert azimuthal equal area and Albers equal area.
Figure 5: Digital elevation model of an area near Fairbanks. The “holes” in the DEM represent areas that could not be unwrapped.
4. TUTORIAL

The exercises described in this chapter provide all the information required to handle and process ASF data using the ASF software tools, starting from reading in the data until the creation of a digital elevation model and the error analysis of the final product.

4.1 READING DATA SETS FROM TAPE

In order to check the status of a tape drive and the functionality of a tape the UNIX command `mt` can be used.

> mt –f <devicename> status

The response from the system should be something similar to the following examples.

**Exabyte:**
- Controller: SCSI
- Device: EXABYTE: EXB-85058HE-00000112
- Status: 0x20266
- Drive type: 8mm(8500) cartridge
- Media: READY, write protected, at BOT

**DLT:**
- Controller: SCSI
- Device: QUANTUM: DLT7000
- Status: 0x20262
- Drive type: DLT
- Media: READY, writable, at BOT

Details about the devices on a particular system can be found in the directory `/dev/rmt`. With the correct device name and the tape drive functional, the data set can be read from the tape.

All level one ASF data is distributed with a product report that contains the basic information about each order.

Order Id: <order ID>
User Name: <user name>
Account Id: <account ID>
Media Id: AFPDTG005462
Media Type: 8-MM HD
Media Format: TAR(CEOS)
Media Sequence: 1 of 1
Media Creation Date: 02-12-2001
Media Deformat Routine: tar xvf /dev/rmt/0 (CEOS_reader)
... 

Level zero products are currently not part of the regular ordering system. Therefore, the shipment report contains very limited information about the data sets such as the list of files on the tape and the media format.

The piece of information essential for reading the tape is the media format (tape format in level zero products). There are two different formats, viz. the tar format and the CEOS format.

A tape stored in TAR format needs to be read using the UNIX command `tar`.

> tar xvf <devicename>

The tar command will list the names of the data sets as they are read off the tape.

A tape stored in CEOS format needs to be read with the `dump_multi_volume` tool.

> dump_multi_volume –s <devicename>

The tool reads the table of contents and prompts for the files to be extracted.

### 4.2 Pre-processing of Data into ASF Internal Format

Depending on the type of data, the pre-processing of the data into ASF internal format is slightly different.

Level zero swath (STF) data need to be converted using the `lz2raw_flywheel` tool.

> lz2raw_flywheel <input> <output>

The input file requires the extension .000 (or whatever the sequence number of the binary data file happens to be). The program assumes that a parameter file with the extension .000.par exists in the same directory.
Level zero CEOS data (frames) are ingested with the tool ceos2raw.

> ceos2raw <input> <output>

The input file requires an extension (either .D or .dat). The program is looking for a leader file with the extension .L or .ldr in the same directory.

Both level zero raw signal data formats need to be processed with the ASF interferometric processor aisp, described in section 4.3, into a complex data format.

For level one CEOS data, the header information has to be stripped from the actual data using the tool trim_slc.

> trim_slc <inSLCfile> <outcpxfile> <new top line> <new left sample>
  [<trim new height> <trim new width>]

The input file is expected to have the extension .dat or .D but does not need to be specified. If the data is not supposed to change in size, the new top line and new left sample have to be set to zero. New height and width values for the output image are optional. The program creates a data descriptor file (.ddr) that can later be used for the visualization of the amplitude or phase image, once the complex image has been converted into polar coordinates.

Data in the CCSD format can be directly processed with the ASF Interferometric SAR processor.

### 4.3 The ASF Interferometric SAR Processor (AISP)

The ASF Interferometric SAR processor (aisp) converts SAR raw signal data into a single look complex image. It handles raw data converted from level zero data as well as data in CCSD format.

The processor works with parts of the data on a patch by patch basis. A patch consists of 4096 lines of data. The number of range samples varies with the beam modes of the various satellites. For ERS imagery with its single standard beam mode the number of range samples is 5616, which is processed to 4800 output range samples.

Various options allow processing subsets of any size and position within the data set. Aisp can process the data sets to a certain Doppler value. For ERS imagery usually the average Doppler of the image pair, stored in an external text file, is used. If the difference in Doppler between the two images is significantly large, average Doppler processing introduces artifacts in the interferogram. In this case, the deskewing of both images to zero Doppler is required. If no Doppler value is given, the images are processed to their individual Doppler values extracted from the metadata.
For processing the slave image, an offset estimated during the coregistration can be applied. This approach avoids an additional resampling step within the coregistration that would lead to a decrease in data quality. The antenna gain values can be corrected if more accurate values are available from calibrating the data.

By default, the azimuth reference function uses a rectangular window to reduce the sidelobes of the data spectrum. Alternatively, a Hamming window can be selected. For troubleshooting, a variety of intermediate results such as range reference function, range migration, range compression, range specs and azimuth compressed patch can be stored. As default, the ASF Interferometric SAR processor calculates an amplitude image for each image.

4.4 GETTING INFORMATION ABOUT DATA

Depending on the type of data the metadata information is stored in different files. Level zero swath (STF) data have a parameter file (.par file) that can be viewed with any text editor. Level zero CEOS data or level one single-look complex (SLC) data have a metadata file that follows the CEOS standard (.ldr/L file). As this leader file is not completely in ASCII format, the metadata tool must be used to view the contents. The metadata contain the following records:

- Data Set Summary Record (contains essential information about data acquisition etc.)
- Map Projection Data Record (is only populated if SAR data are geocoded)
- Platform Position Data Record (contains the state vectors)
- Attitude Data Record
- Radiometric Data Record
- Data Quality Summary Record
- Signal Data Histograms Record
- Processed Data Histograms Record
- Range Spectra Record
- Facility Related Data Record
- Image File Descriptor Record (is actually part of the binary data file)
- Leader File Descriptor Record

When the SAR data are converted from their raw data format, most of the relevant information is stored in the .in and .meta file. General radar parameters used for the processing are kept in the .in file. All relevant metadata extracted from the CEOS leader file or the parameter file (parameters used in geolocating the image, interferometry related parameters, state vectors and extra sensor information) is saved in the metadata file (.meta file).

All images that can be visualized are accompanied by a data descriptor record file (.ddr file). The data descriptor file contains all information about a data set (dimensions, data type, geolocation and map projection) that is required to visualize the SAR data. It can be viewed using the dspddr tool.
4.5 **Having a Look at Data**

All intermediate products are stored in LAS format. Except the complex raw data any intermediate or final result (amplitude images, phase images or elevation data) can be viewed using the tools SARview or XID. The functionality of SARview is described in Appendix B. A detailed description of XID is given in Appendix C.

If there is no amplitude image created by running the aispl, which is the case for level one CEOS data, the complex data need to be converted first into an amplitude image.

The program `c2p` converts a complex image into an amplitude and a phase image.

```
> c2p <input> <output>
```

It expects an extension `.cpx`, which should not be added on the command line. The output file should have the same name as the input file. Otherwise the data descriptor file (.ddr), created by `trim_slc`, has to be copied to the output file base name.

If the amplitude image is just used for visualization, it is advisable to convert the amplitude image that has been created by `c2p` from a floating-point image into a byte image.

```
> convert2bye [-look lx] [-step lx] [-multilook] <infile> <outfile>
```

If no look area or step area are specified, the program assumes ERS imagery that needs to be multilooked with five looks to create approximately square output pixels. Alternatively, metadata file can be given to determine the number of looks. The input file requires the extension .amp.

4.6 **Radiometric Calibration of SAR Imagery**

SAR data are generally calibrated before they are widely distributed. A processor is calibrated when the coefficients required for accurate radiometry have been determined, but an image is calibrated only when those coefficients have been applied.

Calibrating a SAR image is the process of converting a linear amplitude image into a radiometrically calibrated logarithmic power image. The input image is in units of digital numbers (DNs), whereas the output image is in units of $\sigma_0$, which is the ratio (in dB) of the power that comes back from a patch of ground to the power sent to the patch of ground. The `calibrate` tool performs this task.

```
> calibrate <inputName> <outputName>
```
The data values of the output image can only be between 0 and 255, but \( \sigma_0 \) ranges from -25.5 dB to 0 dB. \( \sigma_0 \) values are converted into image DNs by

\[
DN = \sigma_0 \cdot 10 + 255
\]

which can be inverted by

\[
\sigma_0 = \frac{DN - 255}{10}.
\]

This results in the following output.

![Figure 6: Before calibrate (linear amplitude)](image1)

![Figure 7: After calibrate (dB power)](image2)

### 4.7 Resampling

Resampling is one of the most critical issues within the co-registration of data. There are a large variety of resampling methods that differ in complexity and computational efficiency. The higher the level of complexity and accuracy of a resampling method, the larger the impact it causes in terms of processing time. Therefore, resampling is generally a trade off between accuracy and processing time. Also, any resampling degrades resolution.

One approach to the problem is to avoid any additional resampling steps in the processing chain. This saves time and does not degrade the resolution of the data sets. Following this strategy, the co-registration of level zero data determines the offset between the master and slave image for selected patches before the images are processed with the ASF interferometric SAR processor (aisp). This way the resampling of the slave image can be performed during the SAR processing and does not require additional resampling.
For resampling using the `remap` tool, three different interpolation methods have been implemented. The *nearest-neighbor* interpolation takes the nearest input pixel and copies it to the output. It is the fastest resampling method. It does not blur sharp edges, but it cannot shift images by a non-integral number of pixels. Using *bilinear* interpolation for the resampling takes a linear weighted average of four surrounding input points to generate an output point. It is slightly slower than the nearest-neighbor method and allows a sub-pixel shift. However, it tends to blur sharp edges in the input. The resampling using an 8-point *Sinc* function takes a Fourier-weighted average of the nearest 64 input pixels to generate a single output pixel. It is several times slower than the other methods, but allows for fine sub-pixel shifts without blurring sharp edges.

The choice of the interpolation also effects the level of correlation calculated in the coherence. The coherence can improve up to ten percent when more accurate interpolation has been used during the co-registration of master and slave image.

### 4.8 Quality of Coregistration

In order to check the quality of the co-registration, two different measures can be used.

Three matching techniques based on amplitude, coherence and complex radar image values have been implemented. The computationally fastest way to use any matching technique is to perform it in the frequency domain using Fast Fourier Transformations (FFTs). An advantage of the FFT approach is that you can visually check the quality of the match. The `fftMatch` tool, that calculates the initial offset between master and slave image, allows storing the two-dimensional FFT correlation matrix that ideally shows a single peak in the center of the image.

![Figure 8: Bad correlation images come from different images](image1)

![Figure 9: Good correlation images come from similar images](image2)
An alternative approach evaluates the coherence image formed by the co-registered images. The level of coherence represents a direct measure of how well the two corresponding SAR signals from master and slave image match each other. One drawback of this approach is that there are other potential sources such as temporal and baseline decorrelation, atmospheric effects etc. that influence the coherence level as well.

4.9 BASELINE REFINEMENT

Once an interferogram has been generated and the flat-earth term has been removed from the phase, there are still a few extra fringes in the phase image. These extra fringes are the result of an incorrect estimation of the baseline from the state vector describing the orbit. The baseline needs to be refined using control points of known geolocation that should be evenly spread across the entire image. The baseline is refined using the unwrapped phase and seed points generated from the simulated phase derived from a low-resolution reference DEM. This is an iterative process that usually takes a couple of iterations before the solution converges.

The precise orbit state vectors can be obtained from the following institutions:
- German processing and archiving facility (D-PAF)
- Delft Institute for Earth-oriented Space Research (DEOS)
  http://www.deos.tudelft.nl/ers/precorbs/

4.10 PHASE FILTERING

In order to improve the phase unwrapping results, it is advisable to filter the phase before actually unwrapping it. An approach developed by Goldstein and Werner (1998, GRL, 25(21):4035-4038) has been implemented.

The interferogram power spectra are characterized by a “white” component generated by thermal noise and loss of coherence as well as a narrow band component related to fringes. The fringe is determined by the look angle, along-track changes in the baseline and any motion of the scene along the line of sight. The adaptive phase filtering approach by Goldstein and Werner is sensitive to local phase noise and the fringe rate. It performs a segmentation of the interferogram into overlapping rectangular patches. The power spectrum is computed by smoothing the intensity of the two-dimensional FFT. The spatial resolution of the filter adapts to the local phase variation. This way, regions of smooth phase are strongly filtered, whereas regions with high phase variance are weakly filtered.

The method effectively reduces the number of residues, which simplifies the phase unwrapping as well.
4.11 QUALITY ASSESSMENT OF DEMS

For the quality assessment of InSAR DEMs there is a variety of methods. Most commonly, the quality is estimated by comparing the InSAR DEM with another reference DEM or to ground control points.

In addition to these methods, an error map can be created based on the coherence image, the baseline and the unwrapped phase using the eleverr tool. The error map contains a $1\sigma$ error on a pixel by pixel basis in slant range geometry. Problem areas from the phase unwrapping are masked out.

This kind of error map has the advantage that it estimates the error based on the performance of the processing. Furthermore, it provides a continuous error estimate.

Figure 10: Slant range DEM error map

Figure 11: Coherence image used for error estimation
4.12 GEOCODING

Geocoding the image lines it up to a standard map projection, making images easier to put together. It defines the transformation between the local coordinate system and global Cartesian coordinates. An introduction into the various map projections that can be used for the geocoding is given in the next section.

The Universal Transverse Mercator (UTM) projection is one of the most common projections. A projection parameter file for the UTM zone 6 (central Alaska) with projection key utm6 can be created using the projprm tool.

> projprm UTM utm6 met utm.proj -z 6.

With the projection parameter file any LAS format image can be geocode. The geocode tool adds a coordinate system and a map projection to the image.

> geocode -p 100 <input>.meta <input>.img utm.proj utm6 <output>

An geocoding example is shown below.

![Figure 12: Before geocoding (SAR coordinates)](image1)

![Figure 13: After geocoding (UTM coordinates)](image2)
4.13 Map Projections

Map projections are an essential part of geocoding data. They allow the user to combine and analyzes data sets from different sources as they have the same reference. There are two different ways to group map projections.

The first way describes the geometrical figure (cylinder, cone or plane) that has been used for the projection. The cylindrical projection takes a cylinder that has its entire circumference tangent to the Earth’s surface along a great circle (e.g. the equator). The conic projection uses a cone that is tangent to the surface along small circles (e.g. a parallel of latitude). The azimuthal projection projects positions directly to a plane tangent to the Earth’s surface.

The second way distinguishes the projections according to their geometric characteristics. Equidistant projections maintain a constant scale in all directions from one or two standard points. Equal-area projections ensure that equal areas are represented by the same map area regardless of where they occur. Conformal projections have angles that are the same as measured on the Earth’s surface and have meridians that intersect parallels at right angles.

Currently, the ASF software tools support the following map projections.

Geographic projection:
The Geographic projection is the most commonly used projection. Its coordinates are given in arc seconds, which can be converted to latitude and longitude. No further projection parameters need to be defined.

Universal Transverse Mercator projection:
The Universal Transverse Mercator (UTM) projection is a commonly used conformal cylindrical projection. Its central meridian and the equator are straight lines. The scale is constant along any meridian. The central meridian is mapped at a scale of 0.9996. As projection parameters, the latitude and longitude of a point and the zone number need to be defined.

Albers Conic Equal-Area projection:
The Albers Conic Equal-Area projection has parallels that are concentric circular arcs. The meridians are equally spaced. The scale is true along standard parallels, smaller between them and larger outside the parallels. The scale variation along the meridians is required to maintain the equal area constraint. As projection parameters, the North and South standard parallel and the central meridian need to be defined.

Lambert Azimuthal Equal-Area projection:
The Lambert Azimuthal Equal-Area projection has a scale that is true at the center point, decreases in radial direction away from the center and increases with distance perpendicular to the radius. In its polar form (pole is center point), the meridians have straight radii and the parallels are concentric circles. In its oblique form only the central meridian is straight and other meridians and parallels are complex curves. As projection
parameters, the center latitude and longitude as well as the radius of the reference sphere need to be defined.

**Polar Stereographic projection:**
The Stereographic projection is a conformal azimuthal projection that is most commonly used to map polar regions. With the pole as the center point, the meridians have straight radii and the parallels are concentric circles. As projection parameters, the center longitude and the standard parallel need to be defined.

### 4.14 IMAGE MOSAICKING

The area of interest for a project can easily be larger than a single image covers. In this case, several images have to be combined into a mosaic to form the final product. A mosaic, being a cartographic product, must have some map projection, described in the previous section. While mosaicking, the images are automatically calibrated and geocoded before they are combined. This leads to a seamless mosaic.

Depending on the type of input imagery, there are different tools available for the mosaicking:
- Raw CEOS images: > auto_sar_mosaic utm.proj utm6 outMosaic input1 input2
- Geocoded SAR amplitude images: > concat <output> <amp1.img> <amp2.img> ...
- Geocoded DEMs: > concat_dem <output> <dem1.img> <dem2.img> ...
An mosaicking example is shown below.

**Figure 14**: Image 1 (SAR coordinates)

**Figure 15**: Image 2 (SAR coordinates)

**Figure 16**: After geocoding (UTM coordinates)
This page is intentionally blank
5. TOOLS

This chapter provides information on the ASF software tools in an alphabetical order.

For each tool a brief description, its usage and a list of input as well as output files is given. For complex scripts the specific processing flow is laid out. An example of a program output allows the troubleshooting and comparison with the user’s processing results. The cross-references help in finding other related software tools.

There are a couple of conventions used throughout this chapter, mainly to explain the usage of the tools.

1. `<file>` indicates that the software tool is looking for a file name.
2. `[optional]` indicates the optional use of these files or parameters
3. In the PDF version, green highlighted tool names provide cross-link to other tools within this manual
**accum_offset**

**Description:**

*accum_offset* is a program that calculates the accumulative offsets of a chain of images from the metadata. The program orders all input images in ascending latitude and calculates the cumulative offset of each image from the first image. It uses the upper left-hand coordinates for its calculations.

**Usage:**

```
accum_offset <metafile1> <metafile2> [...] 
```

**Input:**

metafile[1-n] – number of metafiles

**Program log:**

```
accum_offset R139153153G3S006.meta R139153154G3S007.meta

<table>
<thead>
<tr>
<th>File</th>
<th>Line</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>R139153153G3S006.meta</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R139153154G3S007.meta</td>
<td>-886</td>
<td>-26</td>
</tr>
</tbody>
</table>
```

Mosaic Image Positions:

```
File: R139153153G3S006.meta Line: 887.95 Sample: 26.95
File: R139153154G3S007.meta Line: 0.56 Sample: 0.44
```

Total number of lines: 10484
Total number of columns: 11026

**used by:**

`auto_swath_mosaic` (p. 45)

**Notes:**
**aisp**

**Description:**

`aisp` is a program that converts a SAR raw signal data (internal format) into a single look complex image and a multilooked amplitude image. The data is processed on a patch basis. After performing a compression and migration in range direction, the data is compressed in azimuth direction as well. Finally, the data is transposed into range line format. The program supports a large number of options indicated in the list below (with default values assumed if the option is not given).

**Usage:**

```
aisp [ options ] <input> <output>
```

**Input:**

- `input` – input file
  - `input.raw`
  - `input.in`
  - `input.fmt`
  - `input.ddr`

**Output:**

- `output` – output file
  - `output.cpx` – single-look-complex float SAR image
  - `output.ddr` – data descriptor for output image
  - `output.in` – parameter file that was used for this run
  - `output.meta` – metadata file containing state vectors, etc.
  - `output_amp.img` – multilooked float SAR amplitude image
  - `output_amp.ddr` – data descriptor for amplitude image.

**Options:**

<table>
<thead>
<tr>
<th>Options</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-l first_line</code></td>
<td>1</td>
<td>First line to process (from 0)</td>
</tr>
<tr>
<td><code>-p patches</code></td>
<td>8</td>
<td>Number of patches to process (@ 4K lines)</td>
</tr>
<tr>
<td><code>-v valid_lines</code></td>
<td>3000</td>
<td>Valid output lines per patch</td>
</tr>
<tr>
<td><code>-s skip_samp</code></td>
<td>0</td>
<td>range samples to skip (of INVALID samps)</td>
</tr>
<tr>
<td><code>-f first_samp</code></td>
<td>0</td>
<td>1st range samp to process (of VALID samps)</td>
</tr>
<tr>
<td><code>-e 1</code></td>
<td>0</td>
<td>remove doppler skew from image (flag).</td>
</tr>
<tr>
<td><code>-n num_samps</code></td>
<td>META</td>
<td>Number of range samples to process (Default is read from metadata)</td>
</tr>
<tr>
<td><code>-r output_res</code></td>
<td>8.0</td>
<td>Desired output azimuth resolution (m)</td>
</tr>
<tr>
<td><code>-debug dbg_flg</code></td>
<td>1</td>
<td>Debug: 1=amplitude,2=ref_fcn,4=rangemig,8=rangecomp,16=rangescens,64=acpatch</td>
</tr>
<tr>
<td><code>-c dfile</code></td>
<td>NO</td>
<td>Read doppler centroid from dfile</td>
</tr>
<tr>
<td><code>-o off_file</code></td>
<td>NO</td>
<td>Read resampling coefficients from off_file</td>
</tr>
<tr>
<td><code>-hamming</code></td>
<td>NO</td>
<td>Use a Hamming window instead of a rectangular one for the azimuth reference function weighting</td>
</tr>
<tr>
<td><code>-m CAL_PARAMS</code></td>
<td>NO</td>
<td>Read the Elevation Angle and Gain vectors from the CAL_PARAMS file to correct for the antenna</td>
</tr>
</tbody>
</table>
**Program log:**

Command line: aisp -p 1 -v 3300 -d 1 -c reg/avedop -l 250818 -log aisp.log -quiet b reg/b_p1

Date: Sat Sep 8 13:02:45 2001
Program: aisp

Processed 1 patch.

Total wall clock time = 120.500000 seconds.

used by:  create_dem (p. 62)

see also:  calc_deltas (p. 49)

Notes:

If input file is ASF internal raw format, first line should read "Found raw, manual file …". However, if the input is a CCSD file the first line should read "Found ASF CCSD file …"
**auto_sar_mosaic**

Description:

`auto_sar_mosaic` is a script that assembles a set of SAR CEOS input files into a large mosaic in LAS format. The script ingests the individual CEOS images with (calibrate) or without (sarin) applying the calibration parameters. The images are geocoded and automatically mosaicked together with concat using the information given in the data descriptor records.

**Usage:**

```
auto_sar_mosaic [-c] [-h] [-p pixel_size] [-i] [-f fill]
<projfile> <projkey> <output_file> <input files...>
```

**Input:**

- input files – SAR CEOS images
- projfile – projection file (from projprm)
- projkey – projection key

**Output:**

- output file – mosaic image file in LAS format

**Options:**

- `-c` – does not calibrate files
- `-h` – creates a color concatenated image (red, green, blue, red...)
- `-p pixel_size` – sets output pixel spacing
- `-i` – mosaics image ids only, do not create an image mosaic
- `-f fill` – sets background of images to `<fill>` value

**Program log:**

```
auto_sar_mosaic utm.proj UTM5 mosaic R139153153G3S006 R139153154G3S007

Working on ####### R139153153G3S006 - R139153153G3S006_geo #######

Ingesting R139153153G3S006 ...

Calibration Comments: Processor artifacts may impact image quality.
ALPHA123 new x-vector: -0.107,0.104,0.989
alpha1, alpha2, alpha3  315.704410 -98.577445 -31.168281
Output-Sigma0-FILE : las_tmp

Substituting hardcoded noise vector sspswb016
Input and output nl=11000, ns=11000
Now Processing Line No = 0
Now Processing Line No = 100
...
Now Processing Line No = 10800
Now Processing Line No = 10900
Wrote 11000 lines of 11000 samples

Calibration is complete!

Doing geocode las_tmp las_tmp utm.proj UTM5 R139153153G3S006_geo
Command line: fit_quadratic tmp.tie tmp.map
```
ASF software tools – Tools

Command line: remap -nearest -quadratic tmp.map -asDDR tmp.ddr las_tmp R139153153G3S006_geo

Working on ###### R139153154G3S007 - R139153154G3S007_geo ######

Ingesting R139153154G3S007 ...
Calibration Comments: Processor artifacts may impact image quality.
ALPHA123 new x-vector: -0.107, 0.104, 0.989
alpha1, alpha2, alpha3  315.676782 -98.577510 -30.772188
Output-Sigma0-FILE : las_tmp

Substituting hardcoded noise vector sspswb016
Input and output nl=11000, ns=11000
Now Processing Line No = 0
Now Processing Line No = 100
... 
Now Processing Line No = 10800
Now Processing Line No = 10900
Wrote 11000 lines of 11000 samples

Calibration is complete!
Total wall clock time = 77.519997 seconds.
Doing geocode las_tmp las_tmp utm.proj UTM5 R139153154G3S007_geo
Command line: fit_quadratic tmp.tie tmp.map
Command line: remap -nearest -quadratic tmp.map -asDDR tmp.ddr las_tmp R139153154G3S007_geo

Combining multiple images into single image with concat

[concat] Output image is mosaic
[concat] Input image 1 is R139153153G3S006_geo
[concat] Input image 2 is R139153154G3S007_geo
[chkwindow-window] Window specified for file mosaic.img is (817 337 14282 14235)
[chkwindow-window] Window specified for file mosaic.img is (1 1 14275 14229)

[concat] Successful completion; Wrote image mosaic

see also: auto_swath_mosaic (p. 45), sarin (p. 119), calibrate (p. 50), geocode (p. 88), concat (p. 56)

Notes:
auto_swath_mosaic

Description:

auto_swath_mosaic is a script that creates a mosaic from two or more non-geocoded SAR images in CEOS format. The script ingests the SAR data with (calibrate) or without (sarim) applying the calibration parameters. It uses accum_offset to determine the offsets between the individual images using the orbit information provided in the metadata file. The images are then mosaicked together with concatm. Optionally, the final output can be resampled.

Usage:    auto_swath_mosaic [ -c ] [ -r sample_size ] <output_file> <input images...>

Input:    input images  –  non-geocoded CEOS images

Output:   output file    –  geocoded mosaic in LAS format

Options: -c           –  no applying of calibration parameters to the imagery
               -r sample_size –  pixel size for resampling the mosaic

Program log:

auto_swath_mosaic R139153153G3S006 R139153154G3S007

Adding image R139153153G3S006 to the list.
Adding image R139153154G3S007 to the list.
Ingesting the following images:

Ingesting image R139153153G3S006
Calibration Comments: Processor artifacts may impact image quality.
ALPHA123 new x-vector: -0.107,0.104,0.989
alpha1, alpha2, alpha3 315.704410 -98.577445 -31.168281
Output-Sigma0-FILE : R139153153G3S006
Substituting hardcoded noise vector sspswb016
Input and output nl=11000, ns=11000
Now Processing Line No = 0
Now Processing Line No = 100
...  Now Processing Line No = 10800
Now Processing Line No = 10900
Wrote 11000 lines of 11000 samples
Calibration is complete!

Ingesting image R139153154G3S007
Calibration Comments: Processor artifacts may impact image quality.
ALPHA123 new x-vector: -0.107,0.104,0.989
alpha1, alpha2, alpha3 315.676782 -98.577510 -30.772188
Output-Sigma0-FILE : R139153154G3S007
Substituting hardcoded noise vector sspswb016
Input and output nl=11000, ns=11000
Now Processing Line No = 0
Calibration is complete!

Total wall clock time = 77.559998 seconds.

Obtaining offset data values...

<table>
<thead>
<tr>
<th>File</th>
<th>Line</th>
<th>Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>R139153153G3S006</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>R139153154G3S007</td>
<td>-886</td>
<td>-26</td>
</tr>
</tbody>
</table>

Mosaic Image Positions:

- File: R139153153G3S006 Line: 887.95 Sample: 26.95
- File: R139153154G3S007 Line: 0.56 Sample: 0.44

Total number of lines: 10484
Total number of columns: 11026

Combining multiple images into single images with concatm

[concatm] Output image is mosaic
[concatm] Input image 0 is R139153153G3S006
[concatm] Input image 1 is R139153154G3S007
[chkwindow-window] Window specified for file mosaic.img is (887 26 9598 11000)
[chkwindow-window] Window specified for file R139153153G3S006.img is (1 1 9598 11000)
[chkwindow-window] Window specified for file R139153154G3S007.img is (2 2 10484 10999)
[chkwindow-window] Window specified for file mosaic.img is (1 1 10484 10999)
[chkwindow-window] Window specified for file R139153154G3S007.img is (2 2 10484 10999)

[upddr-proj] 1049192827 is not a valid GCTP projection
[upddr-proj] The projection coefficients will be compared with a precision of 2 decimal places

[concatm] Concat Manual Successful Completion; Wrote image mosaic

```
see also: auto_sar_mosaic (p. 43), sarin (p. 119), calibrate (p. 50), concatm (p. 58)
```

Notes:
**avg\_in\_dop**

**Description:**

`avg\_in\_dop` is a program that averages the Doppler polynomials in the processing parameter files of the master and slave images.

**Usage:**

```
avg\_in\_dop [ -log <file> ] <infile1.in> <infile2.in> <outfile>
```

**Input:**

- `infile1.in` – processing parameter file of master image
- `infile2.in` – processing parameter file of slave image

**Output:**

- `outfile` – average Doppler parameters

**Options:**

- `-log` – allows the output to be written to a log file

**Program log:**

```
Command line: avg\_in\_dop -log avg\_in\_dop.log a b reg/avedop
Date: Sat Sep 8 12:57:54 2001
Program: avg\_in\_dop

Average: 1.506266e-01 -8.560500e-07 0.000000e+00
```

**used by:**

- `refine\_base` (p. 112), `create\_dem` (p. 62)

**see also:**

- `deramp` (p. 69)

**Notes:**
c2p

Description:

c2p is a program that converts a complex SAR image into an amplitude image and a phase image using the standard formulas given below:

\[
\text{amplitude} = \sqrt{\text{re}^2 + \text{im}^2} \\
\text{phase} = \arctan\left(\frac{\text{im}}{\text{re}}\right)
\]  \hspace{1cm} (5-1)

where \(\text{re}\) is the real part and \(\text{im}\) is imaginary part of the complex number.

Usage:  \hspace{1cm} \text{c2p} <\text{infile}> <\text{outfile}>

Input:  \hspace{1cm} \text{infile} \quad \text{– complex SAR image (.cpx)}

Output:  \hspace{1cm} \text{outfile} \quad \text{– amplitude file (.amp)}
\hspace{2cm} \text{phase file (.phase)}

Program log:

\text{c2p a a}
\hspace{1cm} \text{bytes read} = 0
\hspace{2cm} \ldots
\hspace{3cm} \text{bytes read} = 1363148800

see also:  \hspace{1cm} \text{convert2byte} (p. 60), \text{p2c} (p. 106)

Notes:
calc_deltas

Description:

*calc_deltas* is a program that converts the format of linear regression coefficients. The linear regression coefficients (*fit_line*) are converted to an offset file that can be used as an input for the SAR image processing (*aisp*).

Usage:  

```
calc_deltas <line1> <lineL> numLines <output_deltas> [ -log <file> ]
```

Input:  

- `line1`, `lineL`: regression lines for the first and last patch
- `numlines`: number of SAR lines between first and last patch

Output:  

- `output_deltas`: linear regression parameters

Options:  

- `-log <file>`: allows the output to be written to a log file

Program log:

Command line: calc_deltas reg/line1 reg/lineL 25243 reg/deltas -log calc_deltas.log

Date: Sat Sep 8 13:10:49 2001

Program: calc_deltas

```
u1 = -0.000186 x + -2.663331
v1 = 0.000198 y + 211.640137
u8 = -0.000237 x + -2.785804
v8 = 0.000195 y + 211.595200

delu = 0.000000 x + -0.000005
delv = 0.000000 y + -0.000002
```

used by:  

create_dem (p. 62)

see also:  

*aisp* (p. 41), *fit_line* (p. 84)

Notes:
 ASF software tools – Tools

**calibrate**

**Description:**

*calibrate* is a program that performs the radiometric calibration of SAR imagery. The digital numbers (DN) of a SAR image are converted to backscatter values ($\sigma_0$) using coefficients from the radiometric data record using

$$\sigma_0 = 10 \cdot \log \left[ a_2 \left( \text{DN}^2 - a_1 \cdot n(r) \right) + a_3 \right].$$

(5-2)

where $n(r)$ is the noise versus range function immediately following the coefficients in the record.

Alternatively, the SAR image can be converted to the incidence angle ($\theta$) related backscatter value $\gamma_0$ that is defined as follows:

$$\gamma_0 = \sigma_0 - 10 \cdot \log(\cos \theta).$$

(5-3)

In order to determine the actual $\sigma_0$ value out of the output image, the user needs to perform the following calculation:

$$\sigma_0 = \frac{\text{outByte}}{10} - 25.5$$

(5-4)

**Usage:**

calibrate <input> <output> [ -m ] [ -g ]

**Input:**

input – uncalibrated CEOS SAR image

**Output:**

output – calibrated LAS image (byte format)

**Options:**

- **-m** – allows modification of radiometric calibration coefficients that have been read from the leader file
- **-g** – allows creation of $\gamma_0$ output images based on the incidence angle at each pixel
Program log:

Output-Sigma0-FILE : cal
Input and output nl=7304, ns=4800
Now Processing Line No = 0
Now Processing Line No = 100
Now Processing Line No = 200
...
Now Processing Line No = 7100
Now Processing Line No = 7200
Now Processing Line No = 7300
Wrote 7304 lines of 4800 samples

Calibration is complete!

Notes:
For Radarsat ScanSAR images the calibration vector is noise versus look angle rather than noise versus range.
Do not run this tool on ASF products of beams that have not been calibrated. If you are uncertain of the status of your data check the calibration comments field of the data quality summary record.
ceos2raw

Description:

`ceos2raw` is a program that decodes SAR signal data in CEOS format, specified by the space agencies, into the ASF internal software format. It supports ERS and JERS data as well as all Radarsat single beam modes. Symbolic links are established for the raw data (.D → .raw) and for the leader file (.L → .ldr). This way the import function for the old CCSD format can be used for this operation. The image start time is corrected and state vector information is propagated. Based on state vector information, parameters such as the earth radius as well as spacecraft height and velocity are estimated more accurately. An AISP parameter file and metadata file are created. The data format file contains information such as antenna gain control values.

Usage:  `ceos2raw <input> <output>`

Input:  input  – SAR signal data file in CEOS format (file extension required)
         a.raw  – CEOS raw signal data
         a.ldr  – leader file

Output: output  – SAR signal data file in the internal format
           a.raw  – ASF internal raw data
           a.in  – AISP parameter file
           a.meta  – metadata file
           a.fmt  – data format file

Program log:

`ceos2raw e2_2774.raw b`

Initializing ERS decoder...
IsImaging=1  OnBoardTime=-1304682
formatCount=415110, rfAtten=38
dwp=0.000232  prf=1679.902394
VEXCEL Level-0 CEOS: Shifted by 11.323277 seconds...
Updating for more accurate earth radius (6360566.01),
height (794553.54), and velocity (7543.27).
Converting line 0
...
Converting line 38000
Finished with input file!

used by:  `create_dem` (p. 62)

see also:  `lz2raw_flywheel` (p. 96)
Notes:
**coh**

**Description:**

*coh* is a program that calculates a coherence image for two coregistered SAR images. The formula is

\[
\rho = \frac{\sum_{n} a \times b^*}{\sqrt{\sum_{n} a \times a^* \cdot \sum_{n} b \times b^*}}
\]  

(5-5)

It reads the number of looks from the metadata file unless another step interval or window size is given. The default look area is 5x1 and default step area is 15x3.

**Usage:**

coh [ -look line x sample ] [ -step line x sample ] <In1> <In2_corr> <Out>

**Input:**
in1, in2_corr – master and coregistered slave image

in1.cpx, in2_corr.cpx – SAR data files

in1.meta, in2_corr.meta – metadata files to derive coherence image dimensions

**Output:**

out – coherence image

coh.img – coherence image file

coh.ddr – data descriptor file

**Options:**

-look – look area (defines size of input area equal to one pixel in output image)

-step – step area (defines which data is used to calculate the value in output image)

**Program log:**

coh a.cpx b_corr.cpx coh

Got nLooks from meta files

Calculating interferogram coherence for a.cpx and b_corr.cpx

Input Size => line: 36520 sample: 4800
Step Interval => line: 5 sample: 1
Window Size => line: 15 sample: 3

Coherence : Occurrences : Percent

<table>
<thead>
<tr>
<th>Coherence</th>
<th>Occurrences</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00 -&gt; 0.10 :</td>
<td>00171418</td>
<td>0.676</td>
</tr>
<tr>
<td>0.10 -&gt; 0.20 :</td>
<td>00531322</td>
<td>2.096</td>
</tr>
<tr>
<td>0.20 -&gt; 0.30 :</td>
<td>00976136</td>
<td>3.851</td>
</tr>
<tr>
<td>0.30 -&gt; 0.40 :</td>
<td>01558502</td>
<td>6.149</td>
</tr>
<tr>
<td>0.40 -&gt; 0.50 :</td>
<td>02364628</td>
<td>9.330</td>
</tr>
</tbody>
</table>
0.50 -> 0.60 : 03633822  14.338
0.60 -> 0.70 : 05861890  23.129
0.70 -> 0.80 : 07451066  29.399
0.80 -> 0.90 : 02740719  10.814
0.90 -> 1.00 : 00054497  0.215

Average Coherence: 0.620  (15720564.5 / 25344500) 99.998027

used by:  create_dem (p. 62)

see also:  igram (p. 90)

Notes:
concat

Description:

concat is a program that creates a mosaic of all geocoded input SAR images to produce an output image of an equal area. The overlapping areas between images can either be averaged or filled with information from one of the images. Optionally, a background mask value for the output mosaic can be defined. Additionally, images can be assigned to the color channels red, green and blue.

Usage:  
concat [ -m mask ] [ -v ovrlp ] [ -c ] [ -n ] <outfile> <file1 file2 ...>

Input:  
file[1-n] – number of input files in LAS format

Output:  
outfile – mosaicked output image

Options:  
-m mask – sets background mask value for output mosaic; default value is 0
-v ovrlp – sets overlap pixel options (REPLAC, LEAVE, or AVER); default value is REPLAC
-c – concatenate successive images into successive bands (e.g. first image → red band, second image → green band, third image → blue band, fourth image → red band); works with any number of images
-n – set everything up, but do not copy image data; this is useful to find the offsets between a set of LAS images

Program log:

concat mosaic.img r1_39153_153_geo r1_39153_154_geo

[concat] Output image is mosaic.img
[concat] Input image 1 is r1_39153_153_geo
[concat] Input image 2 is r1_39153_154_geo
[chkwindow-window] Window specified for file mosaic.img is (807 341 5378 10741)
[chkwindow-window] Window specified for file mosaic.img is (1 1 5369 10738)

[concat] Successful completion; Wrote image mosaic.img

used by:  
auto_sar_mosaic (p. 43)

see also:  
concat_dem (p. 57), concatm (p. 58)

Notes:

56
concat_dem

Description:

concat_dem is a program that combines the geocoded input images to create an output image. The algorithm finds the average value of all non-zero input pixels, throws out pixels that differ by more than 60 m from the average and re-averages the remaining values. The input images can be any sort (byte, short, float, long) of single-banded LAS images in any map projection, but they must all be in the same map projection and have the same pixel size.

Usage: concat_dem <out img.ext> <in img1.ext> [ <in img2.ext> [...]]

Input: in img[1-n] – single-banded input DEMs

Output: out img – mosaicked output DEM

Program log:

concat_dem mosaic_dem.img dem1 dem2

Opening image 'dem1'
Opening image 'dem2'
Output DEM has 6185 lines, 11082 samples, and a pixel size of -50.00x50.00 meters.
Processing line 0
Processing line 50
...
Processing line 6100
Processing line 6150

see also: concat (p. 56), concatm (p. 58), geocoded (p. 88)

Notes:
concatm

Description:

concatm is a program that creates an output image by overlaying each input image into the output space at the user specified locations. This program takes manual offsets to determine the relative location of the input images and mosaics the input together. These offsets can be determined by the user or accum_offset as it is used in auto_swath_mosaic. Optionally, the input images can be combined to a three channel RGB image, a mask value for background pixels can be defined as well as the strategy how to combine pixels in overlapping areas.

Usage:  
concatm [-c] [-m mask] [-v overlap] <outfile> nl ns <file1> sl1 ss1 [<file2> sl2 ss2] [ ... ]

Input:

nl  – number of lines in output image
ns  – number of samples in output image
fileX – input file name #X
slX  – starting line in output image to place fileX
ssX – starting samp in output image to place fileX

Output:

outfile – output file name

Options:

-c  – do color concatenation (first image → red; second → green, third → blue, etc.)
-m mask  – mask value for background pixels; default value is 0
-v overlap – processing option for overlapping pixels (REPLAC, LEAVE, or AVER); default is REPLAC

Program log:

concatm mosaic 10484 11026 r1_39153_153 886 26 r1_39153_154 1 1

[concatm] Output image is mosaic
[concatm] Input image 0 is r1_39153_153
[concatm] Input image 1 is r1_39153_154
[chkwindow-window] Window specified for file mosaic.img is (886 26 9599 11000)
[chkwindow-window] Window specified for file r1_39153_153.img is (1 1 9599 11000)
[chkwindow-window] Window specified for file r1_39153_154.img is (1 1 10484 11000)
[chkwindow-window] Window specified for file mosaic.img is (1 1 10484 11000)
[chkwindow-window] Window specified for file r1_39153_154.img is (1 1 10484 11000)
[upddr-proj] 892677995 is not a valid GCTP projection
[The projection coefficients will be compared with a precision of 2 decimal places] 892677995 is not a valid GCTP projection
[upddr-proj] 892677995 is not a valid GCTP projection
[concatm] Concat Manual Successful Completion; Wrote image mosaic
used by:  auto_swath_mosaic (p. 45)

see also:  concat (p. 56), concat_dem (p. 57), accum_offset (p. 40)

Notes:
convert2byte

Description:

convert2byte is a program that converts any ASF image into a viewable byte image. The program extracts the statistics out of the metadata file. It runs stats if the metadata file does not contain any statistics or the statistics were calculated with some mask value. If there is no multilooking involved the program uses the following set of equations for the linear conversion

\[
slope = \frac{255}{in_{\text{max}} - in_{\text{min}}} \\
offset = -\text{slope} \cdot in_{\text{min}} \\
out = \text{slope} \cdot in + \text{offset}
\]  

(5-6)

The multilooking uses the following equation

\[
out = \left( \sum_{l=1}^{\text{lines}} \sum_{s=1}^{\text{samples}} in^2 \right) \cdot \frac{128}{\text{in}_{\text{mean}}} + 0.5
\]  

(5-7)

Usage: convert2byte [ -multilook ] [ -look lxs ] [ -step lxs ] [ -log log_file ] [ -quiet ]

<infile> <outfile>

Input: infile – input file (with extension) with accompanying metadata file

Output: outfile – output file in byte format

Options:
- -multilook – multilook the data as it is converted to byte.
- -look lxs – change number of look lines (l) and samples (s). (-multilook option is implied.)
- -step lxs – change number of step lines (l) and samples (s). (-multilook option is implied.)
- -log log_file – allows the output to be written to a log file
- -quiet – suppresses terminal output to essential.

Program log:

convert2byte r1_25232.amp r1_25232_byte.img

Date: Tue Sep 30 10:59:21 AKDT 2003
Program: convert2byte

60
Converted data to byte: 100% complete.

Total wall clock time = 3.220000 seconds.

see also: stats (p. 123)

Notes:
create_dem

Description:

create_dem is a driver program that runs the entire SAR interferometric processing chain for generating digital elevation models. It includes some measures to interrupt the processing if the intermediate products are of inferior quality. It reports processing errors in the processing log file. This helps to troubleshoot and streamline the processing chain.

For more information about the processing chain, the configuration file and the functionality consult the create_dem manual.

Usage: create_dem [ -c ] <config file>

Input: config file – configuration file containing all necessary processing parameters

Options: -c – initializes the configuration file from minimum to full content

Notes:
create_dem_grid

Description:

`create_dem_grid` is a program that creates a mapping grid which can be used to extract a portion of a DEM to fit a given SAR image. The grid is created by mapping points from the near range and the far range of the SAR image in latitude and longitude, and then Northing and Easting of the DEM projection.

Usage:  
`create_dem_grid [ -log <file> ] <DEM> <SAR> <meta> <out_grid>`

Input:  
DEM – digital elevation model to create a grid upon  
SAR – radar image for which to create the grid  
meta – metadata of the radar image

Output:  
`out_grid` – mapping grid

Options:  
-`log` -- allows the output to be written to a log file

Program log:

Command line: create_dem_grid -log create_dem_grid.log alaska.img sar_byte.img a.meta dem_grid  
Date: Sat Sep 8 13:14:43 2001  
Program: create_dem_grid

Created a grid of 2x20 points

used by:  
`create_dem` (p. 62)

see also:  
`fit_plane` (p. 85), `remap` (p. 114)

Notes:
**ddr2meta**

**Description:**

*ddr2meta* is a program that converts a data descriptor record file (.ddr) and an old metadata file (.meta – version less than 1.1) into a new metadata file (.meta – version larger than 1.1).

**Usage:**

`ddr2meta <las_name> <meta_name>`

**Input:**

*las_name* – old metadata

**Output:**

*meta_name* – new metadata

**Program log:**

`ddr2meta r1_25232_old.meta r1_25232_new.meta`

***Wrote r1_25232_new.meta.meta from r1_25232_old.meta.ddr and r1_25232_old.meta.meta.

**see also:**  *meta2ddr* (p. 100)

**Notes:**
**dem2phase**

**Description:**

*dem2phase* is a program that generates a phase image from a slant range digital elevation model. The wave number $k$ is determined using the wavelength $\lambda$ as

$$ k = \frac{2\pi}{\lambda} \quad (5-8) $$

The phase can be determined using

$$ \text{phase}_{\text{base}} = \frac{\text{SR}(x) \cdot \sin \phi(x)}{2 \cdot k} \quad (5-9) $$

with the slant range SR and incidence angle $\phi$.

The unwrapped phase $\phi_{\text{unw}}$ can be calculated for each sample $x$ in a line

$$ \phi_{\text{unw}}(x) = \frac{h(x)}{\text{phase}_x}\left[ -B_p \cdot \sin \theta(x) - B_N \cdot \cos \theta(x) \right] \quad (5-10) $$

with the elevation $h$, the parallel baseline component $B_p$, the normal baseline component $B_N$ and the "flat earth" look deviation $\theta$.

**Usage:**

```
dem2phase [-log <file>] <sr_DEM> <meta> <base> <phase>
```

**Input:**
- sr_DEM – slant range DEM used to create the phase image
- meta – metadata file of master or slave image
- base – baseline file

**Output:**
- phase – simulated phase image

**Options:**
- -log – allows the output to be written to a log file

**Program log:**

Command line: dem2phase -log dem2phase.log dem_lined ht a.meta base.00 out_dem_phase.phase
Date: Sat Sep 8 13:20:33 2001
Program: dem2phase

Baseline: Normal: 67.828500, delta: 88.450432
Parallel: -25.023166, delta: -38.157228
Temporal: -0.999629 days

Wrote 5280 lines of simulated phase data.
used by:  create_dem (p. 62)

see also:  dem2seeds (p. 67)

Notes:
dem2seeds

Description:

dem2seeds is a program that searches for good seed point locations in a slant range digital elevation model. The algorithm first uses a dense regular grid of points to calculate the local slope. The points are sorted by local slope and the list is searched for the best points in a less dense grid to ensure a good spatial distribution. These seed point locations are used for refining the interferometric baseline. Good seed points are defined as those having minimal local slope.

Usage: dem2seeds <inDEM> <inAmp> <outSeeds> [ -b ] [ -log <file> ]

Input:  
inDEM – slant range digital elevation model
inAmp – amplitude image

Output:  
outSeeds – tie point file with seeds points (x, y and height in meters)

Options:  
-b – seed points will be selected on the basis of maximal local slope
-log – allows the output to be written to a log file

Program log:

Command line: dem2seeds dem_lined.ht sar_byte.img out_dem_seeds -log dem2seeds.log
Date: Sat Sep 8 13:20:38 2001
Program: dem2seeds

Potential seed points: 9801
Final number of seed points: 2832

Seed point distribution:

X X XXX X
XXX X XX
X X XXXX
XX XX X X
X XX XXX
XXXX XXXX
XXXX XXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXX
XX XX XXXXXX
XXX XXXXXX

used by: create_dem (p. 62)

see also: dem2phase (p. 65)
Notes:
deramp

Description:

*deramp* is a program that removes the phase shift in an interferogram caused by the baseline. The interferogram is corrected for the earth curvature by removing the baseline induced phase shift. The flat earth correction that is to be subtracted from the phase is determined by

\[
\phi_{\text{flat}} = \frac{4 \cdot \pi}{\lambda} \cdot (B_P \cdot \cos \theta - B_N \cdot \sin \theta) \tag{5-11}
\]

with the wavelength \( \lambda \), the parallel baseline component \( B_P \), the normal baseline component \( B_N \) and the "flat earth" look deviation \( \theta \).

Usage:  
deramp [ -backward ] [ -log <file> ] <igram> <meta> <base> <out>

Input:  
*igram* – interferogram  
*meta* – metadata file to determine the dimensions of the output image  
*base* – baseline parameters

Output:  
*out* – deramped interferogram

Options:  
- backward – switch to add the baseline induced phase shift  
- log – allows the output to be written to a log file

Program log:

Command line: deramp igram a.meta base.00 igramd -log deramp.log  
Date: Sat Sep 8 13:20:38 2001  
Program: deramp

Baseline:  
Normal: 67.828500, delta: 88.450432  
Parallel: -25.023166, delta: -38.157228  
Temporal: -0.999629 days

Wrote 50688000 bytes of data

used by:  
create_dem (p. 62)

see also:  
ml (p. 104), igram (p. 90), resolve (p. 118)

Notes:
deskew

Description:

*deskew* is a program that removes squint induced skew from ground range images. The program uses the squint angle of an image along with the look angle to determine the amount of parallelogram shift skew that has been introduced in an image due to the Doppler centroid chosen during image processing. It then remaps the image using bilinear interpolation to remove this skew.

Usage:  
deskew <infile> <inmeta> <outfile> <outmeta>

Input:  
infile – input file name (.img, .ddr)  
inmeta – input image metadata

Output:  
outfile – output file name (.img, .ddr)  
outmeta – output image metadata (.meta)

Program log:

deskew r1_33408_gr.amp r1_33408_gr.meta r1_33408_deskewed.amp r1_33408_deskewed.meta

Maximum Skew = 48.997368  
Reading Line 0  
Reading Line 1000  
...  
Reading Line 12000  
Reading Line 13000

see also:  
*sr2gr* (p. 122)

Notes:
deskew_dem

Description:

*deskew_dem* is a program that converts a slant range DEM to ground range and interpolates across areas that could not be unwrapped. This is done using a first order linear approximation to map slant ranges to ground ranges. Optionally, it corrects a SAR image to ground range.

**Usage:**

deskew_dem <inDEMfile> <ceos> <outfile>
[ <inSARfile> ] [ -g ] [ -log <file> ]

**Input:**
inDEMfile – slant range digital elevation model
ceos – metadata file

**Output:**
outfile – ground range digital elevation model

**Options:**
inSARfile – terrain corrected amplitude image based on slant range DEM
-g – only geometric terrain correction will be applied; no radiometric incidence angle normalization; this is the default for an image of extension .coh.
-log – allows the output to be written to a log file

**Program log:**

Command line: deskew_dem elev_sr.ht a.meta elevation.dem -log deskew_dem.log
Date: Sat Sep  8 13:32:57 2001
Program: deskew_dem

DEM in slant range, but will be corrected.
Correcting DEM geometrically.

Wrote 101376000 bytes of data

**used by:**
create_dem (p. 62)

**see also:**
elev (p. 77), reskew_dem (p. 117)

Notes:
diff_las

Description:

diff_las is a program that compares two LAS images. It creates a histogram of the differences between the two images and some general statistics. The input images have to be of the same size and data type.

Usage: diff_las [-m mask_value] [-d difference.ext] <img1.ext> <img2.ext>

Input: img1.ext, img2.ext – input images

Options: -m mask_value – mask value ignored in the calculation of statistics
-d difference.ext – name of difference image

Program log:

diff_las -d qc unwrap.phase out_dem_phase.phase
The master sample fields do not match: 1 vs 101
Comparing line 0...
Comparing line 100...
...
Comparing line 7100...
Comparing line 7200...
Files differ by more than 0.5%!
Relative Error Histogram:
Diff: Frequency:
-1.00  0.0057%
-0.96  0.0029%
-0.92  0.0032%
-0.88  0.0034%
-0.84  0.0046%
-0.80  0.0068%
-0.76  0.0096%
-0.72  0.0100%
-0.68  0.0125%
-0.64  0.0150%
-0.60  0.0162%
-0.56  0.0192%
-0.52  0.0271%
-0.48  0.0347%
-0.44  0.0399%
-0.40  0.0433%
-0.36  0.0521%
-0.32  0.0693%
-0.28  0.1052%
-0.24  0.1498%
-0.20  0.2079%
-0.16  0.2780%
-0.12  0.3599%
-0.08  0.4948%
-0.04  0.8362%
 0.00  1.9184%
 0.04  3.9993%
 0.08  6.1829%
0.12  7.9231%
0.16 13.6444%
0.20 21.2991%
0.24 15.9892%
0.28 11.0730%
0.32  5.5511%
0.36  2.0624%
0.40  1.4686%
0.44  1.1143%
0.48  0.8427%
0.52  0.6590%
0.56  0.5338%
0.60  0.4249%
0.64  0.3968%
0.68  0.4240%
0.72  0.3190%
0.76  0.2356%
0.80  0.3893%
0.84  0.1926%
0.88  0.0293%
0.92  0.0241%
0.96  0.0266%
1.00  0.4692%

Comparing unwrap.phase to out_dem_phase.phase:
Average difference: -35.730184
Standard deviation: 20.272695
RMS difference: 41.080752
Largest Absolute Difference: 522.515869
Smallest Absolute Difference: 0.000008

used by:  create_dem (p. 62)

see also:  stats (p. 123)

Notes:
**dspddr**

**Description:**

dspddr is a program that displays the information content of the data descriptor file, the file that accompanies each LAS image and contains information such as image dimensions, geolocation and map projection information.

**Usage:**

dspddr [-b band] [-f <filename>] <SARfile>

**Input:**
SARfile – LAS file for which data descriptor file is to be displayed

**Options:**
- b band – bands for which information should be displayed
  - f filename – information is redirected to a file

**Program log:**

dspddr elevation.dem

IMAGE NAME: elevation.dem  
NL: 7304  NS: 4800  NB: 1  DTYPE: REAL*4
LAST MODIFIED:         DATE:             TIME:          SYSTEM: ieee-std
PROJ. CODE: (1) UTM  Valid: VALID
ZONE CODE: 6  Valid: VALID
DATUM CODE: 0  Valid: VALID
PROJ. PARM:  Valid: VALID
A: 0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00
B: 0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00
C: 0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00
D: 0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00
E: 0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00
CORNER COOR:  Valid: VALID
ULcorner: 7.30268785423695E+06 5.11722689845029E+05
URcorner: 7.33170463024727E+06 4.18749813363080E+05
LLcorner: 7.16462335453209E+06 4.67668711138715E+05
LRcorner: 7.19393176024055E+06 3.74975635788029E+05
PROJ. DIST: 1.99582724940000E+01 2.03371207189739E+01  Valid: VALID
PROJ. UNITS: meters  Valid: VALID
INCREMENT: 5.00000000000000E+00 1.00000000000000E+00  Valid: VALID
MASTER COOR: 1 1

******* image: elevation.dem band: 1 *******

MINIMUM: 0.00000000000000E+00  Valid: INVALID
MAXIMUM: 0.00000000000000E+00  Valid: INVALID

DATA SOURCE:
SENSOR TYPE:
CAPT. DIRECTION:
DATE:
TIME:

see also:  makeddr (p. 98)
Notes:
**dump_multi_volume**

**Description:**

*dump_multi_volume* is a program that reads CEOS format data from tape. The tape dumping tool allows for downloading all or selected files of a multi-volume from a tape device. All volumes are processed in sequential order. The select option prompts the user for each file to be downloaded.

**Usage:**

dump_multi_volume [ -v ] [ -t ] [ -s ] <devicename>

**Input:**

devicename – LAS file for which data descriptor file is to be displayed

**Options:**

- **-v** – running the program in verbose mode
- **-t** – use alternative extension (.tlr) for trailer files instead of .trl
- **-s** – allows the user to select files to be dumped

**Program log:**

dump_multi_volume -s /dev/rmt/1n

**************************************************************
Multi Volume Tape Dumping Utility
**************************************************************

Rewinding tape drive /dev/rmt/1n

Reading volume description file...
76+0 records in
76+0 records out

Please select files to dump from tape:
select R105275246G6S016.L, yes or no or done:  y
select R105275246G6S016.D, yes or no or done:  n
select R105275246G6S017.L, yes or no or done:  y
select R105275251G6S017.D, yes or no or done:  d

Dumping selected files from tape...
extracting file R105275246G6S016.L
1+9 records in
1+9 records out

Rewinding tape...

**************************************************************
DUMP_MULTI_VOLUME EXIT SUCCESSFUL
**************************************************************

**Notes:**
elev

Description:

*elev* is a program that converts the unwrapped phase into a digital elevation model. For each pixel in the integrated phase image an elevation $h$ is calculated by

$$h = \left( \phi_{\text{unw}} - \phi_{\text{seed}} \right) \cdot h_{\text{baseline}} - B_p \cdot \sin \theta - B_N \cdot \cos \theta + h_{\text{seed}}$$

(5-12)

with the unwrapped phase $\phi_{\text{unw}}$, the seed point phase $\phi_{\text{seed}}$, the baseline induced height $h_{\text{baseline}}$, the parallel baseline component $B_p$, the normal baseline component $B_N$ and the "flat earth" look deviation $\theta$.

The heights are given in meters. Where the phase could not be unwrapped, the height is set to zero.

Usage:  
elev [ -log <file> ] [ -quiet ] <phase> <base> <meta> <outfile> <seed_file>

Input:  
phase – unwrapped phase  
base – refined baseline  
meta – metadata  
seed_file – tie point file with seed points

Output:  
outfile – slant range DEM

Options:  
-log – allows the output to be written to a log file  
-quiet – suppresses the output to the essential

Program log:

Command line: elev -log elev.log -quiet unwrap.phase base.04 a.meta elevation_sr.ht out_dem_seeds  
Date: Date: Sat Sep  8 13:32:37 2001  
Program: elev

Baseline: Normal: 65.089722, delta: 71.721108  
Parallel: -25.521025, delta: -35.261341  
Temporal: 1.000000 days

used by:  
create_dem (p. 62)

see also:  
deskew_dem (p. 71), eleverr (p. 78), escher (p. 80), refine_base (p. 112), resolve (p. 118)

Notes:
eleverr

Description:

`eleverr` is a program that generates a slant range DEM error map from a coherence image and an unwrapped phase file. The height error \( dH \) is determined by

\[
\begin{align*}
    dH &= \sqrt{dH_{\text{init}}^2 + \left( \frac{h_{\text{baseline}}}{-B_N \cdot \cos \theta - B_P \cdot \sin \theta \cdot \sqrt{1 - \rho}} \right)^2} \\
    &= \sqrt{dH_{\text{init}}^2 + \left( \frac{h_{\text{baseline}}}{B_N \cdot \cos \theta - B_P \cdot \sin \theta \cdot \sqrt{1 - \rho}} \right)^2}
\end{align*}
\]  

(5-13)

with the initial height error estimate \( dH_{\text{init}} \), the baseline induced height \( h_{\text{baseline}} \), the parallel baseline component \( B_P \), the normal baseline component \( B_N \), the "flat earth" look deviation \( \theta \) and the coherence \( \rho \).

For pixels that have not been unwrapped according to the mask image the error value is set to –1.0.

Usage:  
`eleverr [-i init_err ] [-log <file>] [-mask <file>] <coherence> <base> <meta> <outfile>`

Input:  
`coherence` – coherence image  
`base` – refined baseline  
`meta` – metadata  
`mask` – mask image containing integration information

Output:  
`outfile` – slant range DEM error map (containing 1\( \sigma \) errors)

Options:  
- `i init_err` – error associated with seed points in meters (default is 0.0 meters)  
- `log` – allows the output to be written to a log file  
- `mask` – use the mask if escher was used for phase unwrapping

Program log:

Command line: `eleverr -log eleverr.log coh.img base.04 a.meta eleverr_sr.ht`
Date: Date: Sat Sep  8 13:32:42 2001
Program: `eleverr`

Baseline:  
Normal: 65.089722, delta: 71.721108  
Parallel: -25.521025, delta: -35.261341  
Temporal: 1.000000 days

Wrote 101376000 bytes of data

used by:  
`create_dem` (p. 62)

78
see also: coh (p. 54), deskew_dem (p. 71), elev (p.77), escher (p. 80)

Notes:
**escher**

**Description:**

*escher* is a program that uses the Goldstein branch-cut phase unwrapping algorithm to unwrap the given $[-\pi, \pi]$ input file into the phase unwrapped output file. The default location for the start of the phase unwrapping is the image center.

**Usage:**

```plaintext
escher <source.phase> <output.phase> [X Y]
```

**Input:**

- `source.phase` – wrapped phase
- `x` – sample location for start of unwrapping
- `y` – line location for start of unwrapping

**Output:**

- `output.phase` – unwrapped phase
- `output.phase.mask` – mask image used for unwrapping

**Program log:**

```plaintext
escher escher_in unwrap_dem

escher: begin unwrapping phase...
loading wrapped phase...
wrapped phase data loaded...
- grounded border
starting to make mask...
  - made mask

doSstats(): after makeMask():

    35059200 pixels
    34937942 unknown  99.654 %
        0 unwrapped  0.000 %
        84955 residues  0.242 %
->  42480 +residues  0.121 %
->  42475 -residues  0.121 %
  36303 grounds   0.104 %
        0 in tree  0.000 %
        0 cuts  0.000 %

starting branch cut scan [cutMask()]
  - finished branch cutting
doSstats(): after cutMask():

    35059200 pixels
    34579468 unknown  98.632 %
        0 unwrapped  0.000 %
        84955 residues  0.242 %
->  42480 +residues  0.121 %
->  42475 -residues  0.121 %
  84431 grounds   0.241 %
        0 in tree  0.000 %
        443684 cuts  1.266 %
```
checkSeed() finished

starting integratePhase() at (2400, 3652)

from seed point, started out by going up...
integratePhase() unwrapped 34318701 pixels...

doStats(): after integratePhase():

<table>
<thead>
<tr>
<th>Pixels</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>35059200 pixels</td>
<td></td>
</tr>
<tr>
<td>260767 unknown</td>
<td>0.744 %</td>
</tr>
<tr>
<td>34318701 unwrapped</td>
<td>97.888 %</td>
</tr>
<tr>
<td>84956 residues</td>
<td>0.242 %</td>
</tr>
<tr>
<td>42480 +residues</td>
<td>0.121 %</td>
</tr>
<tr>
<td>42475 -residues</td>
<td>0.121 %</td>
</tr>
<tr>
<td>84431 grounds</td>
<td>0.241 %</td>
</tr>
<tr>
<td>0 in tree</td>
<td>0.000 %</td>
</tr>
<tr>
<td>443684 cuts</td>
<td>1.266 %</td>
</tr>
</tbody>
</table>

Zeroing out un-unwrapped phase

Zero'd out un-unwrapped phase
saved a mask array to the file unwrap_dem.phase.mask ...
saved unwrapped phase...

used by: create_dem (p. 62)

see also: deramp (p. 69), phase_filter (p. 107), zeroify (p. 127)

Notes:
The algorithm is originally described in
fftMatch

Description:

*fftMatch* is a program that calculates the offset parameters between two SAR images by correlating the images in the frequency domain using a fast Fourier transformation. The matching process lines up the two images to a single pixel. Optionally, the calculated shifts in x and y direction with the respective certainty level as well as a correlation image can be stored.

**Usage:**

```
fftMatch [ <matchFile> ] [ <corrImg> ] [ -log <file> ] [ -quiet ]
<image 1> <image 2>
```

**Input:**
image 1, image 2 – master and slave image patches

**Options:**
- matchFile – parameter file with shifts and certainty
- corrImg – correlation image to check correlation quality
- -log – allows the output to be written to a log file
- -quiet – suppresses the output to the essential

**Program log:**

Command line: fftMatch -log fftMatch.log -quiet sar_byte dem_trimsim.img dem_corr
Date: Sat Sep  8 13:19:57 2001
Program: fftMatch

Offset slave image: dx = 4.886902, dy = 4.633911
Certainty: 79.521935%

**used by:**
resolve (p. 118), create_dem (p. 62)

**see also:**
fico (p. 83)

**Notes:**
**fico**

**Description:**

*fico* is a program that performs the fine coregistration on a sub-pixel level. The correlation technique uses an approach based on the coherence values. The quality of the offset points is evaluated by comparing forward and backward correlation. The grid size of the correlation points can be increased. This will lead to a better defined offset grid at the expense of processing time. Optionally, a complex FFT approach can be used for the correlation. This can lead to a significant improvement in the coherence level.

**Usage:**

```
fico <file1> <file2> <control> <out>
    [ grid resolution ] [ -f ] [ -log <file> ] [ -quiet ]
```

**Input:**

- file 1, file 2 — master and slave image
- control — parameter file

**Output:**

- out — grid of offset points

**Options:**

- grid resolution — resolution of the correlation grid
- -f — switch for complex FFT correlation
- -log — allows the output to be written to a log file
- -quiet — suppresses the output to the essential

**Program log:**

Command line: fico reg/a_p1 reg/b_p1 reg/ctrl1 reg/fico1 20 -f -log fico.log -quiet
Date: Sat Sep 8 13:04:51 2001
Program: fico

- Using Complex FFT instead of coherence for matching
- Fico attempted 400 correlations, 361 succeeded.

**used by:**

create_dem (p. 62)

**see also:**

fit_line (p. 84), fit_plane (p. 85), fit_warp (p. 87), resolve (p. 118)

**Notes:**

If fico fails when using the default method (based on coherence values), the complex FFT method (-f flag) should be tried and/or the grid size should be increased.
**fit_line**

Description:

*fit_line* is a program that performs a least-squares regression on the offset grid points calculated during the forward and backward correlation in the fine coregistration (*fico*). From the weighted linear first-order least-squares fit a regression line is stored that serves as input to the creation of an offset file (*calc_deltas*) for SAR image processing (*aisp*).

**Usage:**

```bash
fit_line <in> <out> [ -log <file> ] [ -quiet ]
```

**Input:**
in – input file consisting of the forward and backward correlation offset grid points

**Output:**
out – output file

regression line contains the scale and offset in x and y direction as well as the number of points

**Options:**
- log – allows the output to be written to a log file
- quiet – suppresses the output to the essential

**Program log:**

Command line: fit_line reg/fico1 reg/line1 -log fit_line.log -quiet
Date: Sat Sep 8 13:05:47 2001
Program: fit_line

**Fit_line with 361 points:**

```
<img 2>.x=<img 1>.x+ -0.0001874163830507*<img 1>.x+ -2.6588382537869868
<img 2>.y=<img 1>.y+  0.0001973695569021*<img 1>.x+211.6336694417217400
```

x linear, x offset, y linear, y offset:

-0.001858126201125 -2.6633313686204518 0.0001975032346164 211.6401400610922400 289

**used by:**
create_dem (p. 62)

**see also:**
aisp (p. 41), calc_deltas (p. 49), fico (p. 83), fit_plane (p. 85), fit_quadratic (p. 86), fit_warp (p. 87)

**Notes:**
**fit_plane**

**Description:**

*fit_plane* is a program that calculates the transformation parameters needed for resampling the slave image. The transformation coefficients are calculated using a linear least-squares fit. Points are removed according to the respective deviation from the best fit until the fraction of points to be kept is reached.

**Usage:**

`fit_plane <fico> <matrix_file> [ fraction ] [ -log <file> ]`

**Input:**

fico  – correlation points produced by the fine coregistration

**Output:**

matrix_file  – transformation parameter with three coefficients for both x- and y direction

**Options:**

fraction  – fraction of points to get kept (default value is 0.8)

-log  – allows the output to be written to a log file

**Program log:**

Command line: fit_plane dem_grid dem_plane k 1.0 -log fit_plane.log
Date: Sat Sep 8 13:14:43 2001
Program: fit_plane

Number of grid points: 40

\[
\begin{align*}
<x_{\text{out}}>&= -2.9621821329099118*x_{\text{in}}+ -0.4483974384262257*y_{\text{in}}+72448.0750002813730000 \\
<y_{\text{out}}>&= -0.4441838194826470*x_{\text{in}}+ 2.9899737280179162*y_{\text{in}}+ -27118.289125532140000
\end{align*}
\]

Number of points after regression: 40

Coefficients:

\[
\begin{align*}
<x_{\text{out}}>&= -2.9621821329099118*x_{\text{in}}+ -0.4483974384262257*y_{\text{in}}+72448.0750002813730000 \\
<y_{\text{out}}>&= -0.4441838194826470*x_{\text{in}}+ 2.9899737280179162*y_{\text{in}}+ -27118.289125532140000
\end{align*}
\]

**used by:**  
create_dem  (p. 62)

**see also:**  
create_dem_grid  (p. 63),  
fico  (p. 83),  
fit_line  (p. 84),  
fit_quadratic  (p. 86),  
fit_warp  (p. 87),  
remap  (p. 114)

**Notes:**
**fit_quadratic**

**Description:**

*fit_quadratic* is a program that fits a least-squares planar quadratic mapping to a given SAR image. The function fit is of the form

\[
A + B \cdot x + C \cdot y + D \cdot x^2 + E \cdot x \cdot y + F \cdot y^2
\]

for both the x and y coordinates in forward and reverse direction.

**Usage:**

\texttt{fit\_quadratic <input> <output>}

**Input:**

\texttt{input} – tie point file

**Output:**

\texttt{output} – quadratic coefficients file (compatible to the \texttt{--quadratic} option of \texttt{remap})

**Program log:**

Command line: fit\_quadratic -log fit\_quadratic.log tmp.tie tmp.map

Date: Fri Oct 26 16:38:36 2001

Program: fit\_quadratic

Read 81 tie points.

**used by:**

geocode (p.88)

**see also:**

fit\_line (p. 84), fit\_plane (p. 85), fit\_warp (p. 87), remap (p. 114)

**Notes:**
fit_warp

Description:

fit_warp is a program that calculates horizontal and vertical offset grid images. It combines the correlation points derived by fico and weights them into horizontal and vertical transformation images that represent non-linear shifts. This technique is particularly useful for moving or deforming surfaces such as glaciers and ice sheets. The ddr file is used to derive the image dimensions for the horizontal and vertical warp images.

Usage: fit_warp <fico> <image 2> <warp_image>

Input:
- fico – correlation points produced by the fine coregistration
- image2 – slave image

Output:
- warp_image – warp image
  - warp_image.horiz – horizontal warp image
  - warp_image.vert – vertical warp image
  - warp_image.ddr – data descriptor file

Program log:

fit_warp reg/fico b reg/warp_image
294 points read
Output warping images will be 2283 lines by 300 samples
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................
............................................................................................................................................................................

used by: create_dem (p. 62)

see also: fico (p. 83), fit_line (p. 84), fit_plane (p. 85), fit_quadratic (p. 86), remap (p. 114)

Notes:
geocode

Description:

geocode is a program that geocodes a SAR image or DEM into a map projection at a given pixel size. In a first step, a grid of 9x9 points is created, for which the latitude and longitude are determined. The grid point are transformed in the output map projection and the corner coordinates for the new data descriptor record file are calculated. A least-squares planar quadratic mapping function (fit_quadratic) is fitted to transform the image that is remapped (remap) into the map projection. The output image can be resized with various windowing options.

Usage: geocode [ -h ht ] [ -p pix ] [ [-o|-l] N S E W ] [ -i lat lon nl ns ] [-r [near | bilinear | sinc | 'kernel x y' ] ] [-background fill ] [ -x <file> ] <in_meta> <in_img> <projfile> <projkey> <outfile>

Input: in_meta – SAR image metadata or DEM ddr file
in_img – SAR image or DEM to be geocoded
projfile – projection definition file
projkey – projection key name

Output: outfile – geocoded product

Options:

-p pix – pixel size
-h ht – average input elevation to ht meters
-r – resampling options (default is near)
near – nearest neighbor interpolation
bilinear – bilinear interpolation (default value)
sinc – sinc kernel for interpolation
kernel x y – uniform kernel of variable size → low pass filtering

-background fill – background color to the value ‘fill’ (default is zero)
-l N S E W – projection coordinates:
N Y projection coordinate of top edge
S Y projection coordinate of bottom edge
E X projection coordinate of right edge
W X projection coordinate of left edge

-o N S E W – N S are latitude and E W are longitude
N Latitude of north edge of output
S Latitude of south edge of output
E Longitude of east edge of output
W Longitude of west edge of output
-i lat lon nl ns – nl x ns output image with
   lat   Latitude of center of output
   lon   Longitude of center of output
-x – allows the output to be written to a log file

Program log:

Command line: geocode -p 20 -x geocode.log elevation.meta elevation.dem albers.proj albers dem.img
Date: Sun Oct 28 19:13:49 AST 2001
Program: geocode

   Input average output elevation = 0.000000
   Setting output pixel size to 20.00 meters
   Output projection key: ALBERS
   WARNING: Unrecognized map projection code 3 passed to
      geocode/proj.c:write_proj_meta!
      NOT creating .meta file...

Command line: fit_quadratic -log 57_22590_2917.log tmp.tie tmp.map
Date: Sun Oct 28 19:13:49 AST 2001
Program: fit_quadratic

   Read 81 tie points.

Command line: remap elevation.dem dem.img -near -quadratic tmp.map -asDDR tmp.ddr -log
57_22590_2917.log
Date: Sun Oct 28 19:13:49 AST 2001
Program: remap

   Output dimensions: 5843x5465 (LxS).
   Creating image block file remap_215308_0.blk...
   Done creating image block file
   Output Pixel type/size: Floating point pixels (32-bit IEEE).
   Sampling Type: Nearest-neighbor.

see also: fit_quadratic (p. 86), projprm (p. 108), remap (p. 114)

Notes:
igram

Description:

igram is a program that generates an interferogram from two coregistered complex SAR images.
The generation of an interferogram is the result of the complex multiplication of the master and the slave image, i.e. the master image is multiplied with the complex conjugate of the slave image. The resulting interferogram is not deramped yet. Therefore, the phase values are still not corrected for earth curvature. For the interferogram an amplitude image as well as a phase image is created.

Usage:  
igram <a> <b_corr> <igram>

Input:  
a, b_corr – master and coregistered slave image

Output:  
igram – interferogram
igram.amp – amplitude image
igram.phase – phase image
igram.ddr – data descriptor file

Program log:

igram: starting igram
Read chunk 1, 524280 total bytes read
...
Read chunk 2675, 1402368000 total bytes read

Warning, file a is shorter than b_corr!
Output will be truncated to the length of a

igram: Ends successfully

used by:  
create_dem (p. 62)

see also:  
coh (p. 54), deramp (p. 69)

Notes:
**las_op**

**Description:**

*las_op* is a program that calculates an output image based on some mathematical function of the input image (or images). This program works with any type of input data (byte, short, long or float).

**Usage:**

`las_op [ -log <file> ] <output.ext> "expression"`

```
<image A.ext> [ <image B.ext> [...]]
```

**Input:**

- *image A.ext* – input image A in LAS format
- *image B.ext* – input image B in LAS format (optional)
- *expression* – expression parameter is a combination of
  - constants (numbers such as 5 or 1.2345),
  - variables (a, b, c that represent input images)
  - operators (addition \([3=2+1]\), subtraction \([15=16-1]\), multiplication \([6=2*3]\), division \([4.5=9/2]\), modulus \([0.56=3.56%1]\), exponentiation \([9=3^2]\)).

These parts are combined in the expression in ordinary algebraic infix notation, with optional parentheses. The operator precedence is: exponentiation left to right, then \(*\ /\ and \%\ left to right, and finally \(+\ and \-\).

**Examples:**

- `las_op out.img "100+a" in.img` – creates out.img, whose pixels are 100 brighter than in.img
- `las_op out.img "a+b" in1.img in2.img` – creates out.img, whose pixels are the sum of in1.img and in2.img’s pixels.
- `las_op out.img "a^0.5" in.img` – creates out.img, whose pixels are the square root of in.img’s pixels.

**Output:**

`output.ext` – output image in LAS format

**Options:**

- `-log` – allows the output to be written to a log file

**used by:**

`create_dem` (p. 62)
Notes:
las2geotiff

Description:

*las2geotiff* is a program converts a LAS image into GEOTIFF format. The conversion supports a large number of map projections.

Usage: las2geotiff <inLAS> <outGEOTIFF>

Input: inLAS – input image in LAS format

Output: outGEOTIFF – output image in GEOTIFF format

Program log:

las2geotiff e1_34578 e1_34578

see also: las2jpeg (p. 94), las2ppm (p. 95)

Notes:
### las2jpeg

**Description:**

*las2jpeg* is a program that converts an image from LAS format to JPEG format. For the conversion either built-in palette files are used or the one specified by the user. The input image can have one band or three bands.

**Usage:**

```
las2jpeg [ -m <mask> ] [ -p <palfile> ] <infile> <outfile>
```

**Input:**

*infile* – image in LAS format (one band or three bands)

**Output:**

*outfile* – byte image in JPEG format

**Options:**

- `-m mask` – unwrapping mask image
  This option allows color coding the unwrapping bits: white for non-integrated, black for grounding, purple for branch cut, red and green for singularities, and yellow for integrated.

- `-p palfile` – user defined palette file
  This option is useful for colorizing elevation files. A palette file is an ASCII file, where an entry consists of palette index and three color values in the RGB color space separated by white spaces. All values are in the range between 0 and 255.

**see also:** *las2ppm* (p. 95), *las2geotiff* (p. 93)

**Notes:**
las2ppm

Description:

*las2ppm* is a program that converts a byte image from LAS format to PPM format. For the conversion either built-in palette files are used or the one specified by the user.

**Usage:**

```bash
las2ppm [ -m <mask> ] [ -p <palfile> ] <infile> <outfile>
```

**Input:**

- `infile` – byte image in LAS format

**Output:**

- `outfile` – byte image in PPM format

**Options:**

- `-m mask` – unwrapping mask image
  
  This option allows color coding the unwrapping bits: white for non-integrated, black for grounding, purple for branch cut, red and green for singularities, and yellow for integrated.

- `-p palfile` – user defined palette file
  
  This option is useful for colorizing elevation files. A palette file is an ASCII file, where an entry consists of palette index and three color values in the RGB color space separated by white spaces. All values are in the range between 0 and 255.

**used by:**

- create_dem (p. 62)

**see also:**

- las2jpeg (p. 94), las2geotiff (p. 93)

**Notes:**
lz2raw_flywheel

Description:

lz2raw_flywheel is a program that decodes SAR signal data in level zero STF format into the ASF internal software format. It support ERS and JERS data as well as all Radarsat single beam modes.

The image start time is corrected and state vector information is propagated. Based on state vector information, parameters such as the earth radius as well as spacecraft height and velocity are estimated more accurately. A parameter file and a metadata file is created. The data format file contains information such as antenna gain control values. For ERS data, missing lines are found using a length 25 flywheel of the line counters. Gaps are identified and filled with lines of zeros.

Usage:

```
lz2raw_flywheel [ -lat <lower> <upper> ] [ -log <file> ] [ -quiet ] [ -prc path ]
<input> <output>
```

Input:  
input – SAR signal data file in level zero format
a.000 – binary complex data file
a.000.chop – index file for creating CEOS frames (not used)
a.000.par – parameter file of a.000 file
a.log – level zero processing log file (not used)
a.par – overall parameter file (not used)

Output:  
output – SAR signal data file in the internal format
a.raw – ASF internal raw data
a.in – parameter file
a.meta – metadata file
a.fmt – data format file
a.replica – reference function file (applies only to Radarsat data)

Options:  
-lat lower upper – allows for latitude constraints
-log – allows the output to be written to a log file
-quiet – suppresses the output to the essential
-prc – identifies the path where the precise orbit information is stored

Program log:

Command line: lz2raw_flywheel -prc PRC/ERS2 -log 329_23363_3690.log -quiet e2_3690.000 a
Date: Thu Nov  1 17:50:56 2001
Program: lz2raw_flywheel

Look-On Lost at Format 462770
Lock-On Re-established at Format 462799 (1st good format is 462774)
Writing 3 lines of zero data
Writing 25 lines of saved data
Hit end of file looking for aux frame (curFrame = 14299070)
Wrote 467659 lines of raw signal data.

used by:  create_dem (p. 62)

see also: ceos2raw (p. 52)

Notes:
makeddr

Description:

*makeddr* is a program that creates a data descriptor file

Usage:  
makeddr [ -p <projection> <projection parameters> <uln> <ule>  
<lrn> <lre> <pdist> ] [ -d <datum code> ] [ -log <file> ]  
<filename> <nl> <ns> <type>

Input:  
filename  – image for which a data descriptor file needs to be created  
nl  – number of lines  
ns  – number of samples  
type  – # of type  
1 –byte  unsigned 8-bit integer data  
2 –short  signed 16-bit integer data  
3 –long  signed 32-bit integer data  
4 –float  IEEE 32-bit single-precision floating-point data  
5 –double  IEEE 64-bit double-precision floating-point data

Output:  
filename.ddr

Options:  
-p  – adding map projection (specified below)  
projection  – map projection  
GEOGRAPH (0 arguments): nothing  
PLSTEREO  (2 arguments): Standard Parallel, Central Meridian (degrees)  
LAMAZEQA (3 arguments): Central Latitude, Central Longitude (degrees), Radius of reference sphere (meters)  
ALBERS  (3 arguments): 1st Standard Parallel, 2nd Standard Parallel, Central Meridian (degrees)  
UTM  (1 argument): Zone  
uln  – Upper Left Northing Coordinate (meters)  
ule  – Upper Left Easting Coordinate (meters)  
lrn  – Lower Right Northing Coordinate (meters)  
lre  – Lower Right Easting Coordinate (meters)  
pdist  – Projection Distance (meters per pixel)  
-d  – uses a pre-defined datum code (for details see man page)  
-log  – allows the output to be written to a log file
Program log:

makeddr dem_big.dem 7304 5200 -float

Creating base ddr:
  Name: dem_big.dem
  Number of Lines: 7304
  Number of Samples: 5200
  Data type: float

Finished making .ddr file.

used by: create_dem (p. 62)

see also: dspddr (p. 74)

Notes:
**meta2ddr**

**Description:**

`meta2ddr` is a program that converts a new metadata file (.meta – version larger than 1.1) into an old metadata file (.meta – version less than 1.1) and a data descriptor record file (.ddr).

**Usage:**

```
meta2ddr <meta_name> <las_name>
```

**Input:**

- `meta_name` – new metadata

**Output:**

- `las_name` – old metadata

**Options:**

**Program log:**

```
meta2ddr r1_25232_new.meta r1_25232_old.meta

***Wrote r1_25232_old.meta.ddr and r1_25232_old.meta.meta from r1_25232_new.meta.meta.
```

**see also:**

- `ddr2meta` (p. 64)

**Notes:**
metadata

Description:

metadata is a program that retrieves metadata out of the CEOS metadata structure.

Usage:     metadata rectypes <input> [-f]

Input:     rectypes  – record type to be extracted
            u  Data Set Summary Record (dssr)
            m  Map Projection Data Record (mpdr)
            l  Platform Position Data Record (ppdr)
            a  Attitude Data Record (atdr)
            r  Radiometric Data Record (raddr)
            q  Data Quality Summary Record (dqs
            h  Signal Data Histograms Record (dhr)
            p  Processed Data Histograms Record (pdhr)
            s  Range Spectra Record (rsr)
            f  Facility Related Data Record (facdr)
            I  Image File Descriptor Record (ifiledr)
            b  Leader File Descriptor Record (fdr)

input      – SAR image (basename)
-f         – information is redirected to a file with corresponding extension

Program log:

metadata uli e1_24222

*********** begin of Dataset Summary record ***************

DSS SEQ NUM            1
SAR CHNL INDTR         1024
SCENE INDICATOR
SCENE DESIGNATOR       ORBIT=24222-FRAME=2279
INPT SCN CTR TIME      19960302211219864
ASC/DESCENDING         DESCENDING
LAT @ SCN CTR          0.0000000
LONG @ SCN CTR         0.0000000
SCN CTR HEADING        0.0000000
ELLIP DESIGNATOR       GEM6
ELLIP SEMIMAJOR        6378.1440000
ELLIP SEMIMINOR        6356.7590000
EARTH MASS             3.9860045
...  
MISSION ID              ERS1
SENSOR ID               SAR-C-HR-IM-VV
ORBIT NUMBER            24222
PLAT LAT @ NADIR        64.472
PLAT LONG @ NADIR      -142.297
PLAT HEADING            0.000
SNSR CLK ANGLE          90.000
INCIDENCE ANGLE         23.119
RADAR FREQUENCY         5.303
**ASF software tools – Tools**

**Dataset Summary record**

- **RDR WAVELENGTH**: 0.0565646
- **ALT DOPPLER FREQ 1**: 405.3558200
- **ALT DOPPLER FREQ 2**: 0.0002367
- **ALT DOPPLER FREQ 3**: 0.0000000
- **CRT DOPPLER FREQ 1**: 405.3055400
- **CRT DOPPLER FREQ 2**: -0.0064195
- **CRT DOPPLER FREQ 3**: 0.0000000
- **TIME DIRECT RNG**: INCREASE
- **TIME DIRECT AZI**: DECREASE
- **ALT DOPPLER RATE 1**: -2122.8774000
- **ALT DOPPLER RATE 2**: -0.0000089
- **ALT DOPPLER RATE 3**: 0.0000000
- **CRT DOPPLER RATE 1**: -2122.8621000
- **CRT DOPPLER RATE 2**: 0.0211185
- **CRT DOPPLER RATE 3**: -0.0000002

---

**Platform Position record**

- **ORBITAL ELMTS DESG**: EARTH CENTERED ROTATING
- **NUM DATA PTS**: 5
- **YR OF DATA PT**: 1996
- **MON OF DATA PT**: 3
- **DAY OF DATA PT**: 2
- **DAY IN THE YR (GMT)**: 62
- **SECS OF DAY (GMT) OF DATA**: 76328.550000000003000
- **INTRVL BTWN DATA PTS**: 5.657226563198492
- **REF COORD SYSTEM**: EARTH CENTERED ROTATING
- **GREENWICH MEAN HR ANGLE**: 118.967553689290100

---

**Image File Descriptor record**

- **Number of data records**: 38013
- **Record Length**: 11644
- **Bits per sample**: 16
- **Samples per data group**: 1
- **Bytes per group**: 2
- **Justification & order of samples**: 
- **SAR channels**: 1
- **Lines per data set**: 38013
- **Left border pixels per line**: 0
- **Total number of data groups**: 5616
- **Right border pixels per line**: 0
- **Top border lines**: 0
Bottom border lines = 0
Interleave indicator = BSQ
Physical records per line = 1
Physical records per multi-line channel = 0
Bytes of prefix data per record = 412
Bytes of SAR data per record = 11232
Bytes of suffix data per record = 0
Suffix/Prefix repeat flag =
SAR data format identifier = COMPLEX SIGNED INTEGER*2
SAR data format code = CIS2
Left fill bits per pixel = 0
Right fill bits per pixel = 0
Maximum Data Value = 65535
******* end of Image File Descriptor record ***********

Notes:
ml

Description:

`ml` is a program that multilooks an interferogram using coherent summation. The multilooking of an interferogram results in an output that has square pixels. Furthermore, it allows the reduction of speckle and noise by applying a low-pass filter to the image. Finally, an RGB representation of the interferometric phase with the amplitude as background is stored in a LAS image. Optionally, the amplitude image can be removed from the background.

Usage:  

```
ml [ -l lxs ] [ -s lxs ] [ -n <metafile> ] [ -a ] [ -x <logfile> ] <interferogram> <ml>
```

Input: interferogram – interferogram to be multilooked

Output: ml – multilooked output file

- `ml.amp` – multilooked amplitude image
- `ml.phase` – multilooked phase image
- `ml.ddr` – data descriptor file for amplitude/phase image
- `ml_las.img` – interferometric phase stored in an RGB image
- `ml_las.ddr` – data descriptor file for the RGB image

Options:

- `-l` – look area (defines size of input area equal to one pixel in output image)
- `-s` – step area (defines which data is used to calculate the value in output image)
- `-n` – metadata file to set the number of looks (default is 5x1)
- `-a` – switch for removing the amplitude from the multilooked interferogram
- `-x` – allows the output to be written to a log file

Program log:

```
ml -a -n a.meta -x ml.log igram igram_ml
```

Date: Sun Oct 28 21:23:08 2001
Program: ml

- Input is 26400 lines by 4800 samples
- Output is 5280 lines by 4800 samples
- Wrote 101376000 bytes of data

used by: create_dem (p. 62)

see also: deramp (p. 69), igram (p. 90)
Notes:
p2c

Description:

*p2c* is a program that converts an image in polar coordinates, i.e. amplitude and phase, into a complex format with a real and an imaginary part.

It uses the following formula:

\[
\begin{align*}
re &= \text{amplitude} \cdot \cos(\text{phase}) \\
\text{im} &= \text{amplitude} \cdot \sin(\text{phase})
\end{align*}
\]  

Usage:  
p2c <in> <out>

Input:  
\text{in} \quad \text{– amplitude image (.amp) and phase image (.phase)}

Output:  
\text{out} \quad \text{– complex image (.cpx)}

Program log:

```
 p2c r1_33408 r1_33408
   bytes read = 0
   bytes read = 52428800
   ...
   bytes read = 1939865600
   bytes read = 1992294400
```

see also:  
c2p (p. 48)

Notes:
**phase_filter**

**Description:**

*phase_filter* is a program that applies the Goldstein phase filter to an interferogram. The filter raises the fast Fourier transformed phase in the frequency domain to some power and transforms the scaled image back into the time domain. This way the signal (topography or motion) in a phase image is preferentially amplified over the noise (decorrelation).

**Usage:**

```bash
phase_filter [ -log <file> ] <in> strength <out>
```

**Input:**

- `in` – unfiltered phase image
- `strength` – filter strength (experience value: 1.6)

**Output:**

- `out` – filtered phase image

**Options:**

- `-log` – allows the output to be written to a log file

**Program log:**

```
Command line: phase_filter -log phase_filter.log ml.phase 1.6 filtered_phase
Date: Sun Oct 28 21:28:36 AST 2001
Program: phase_filter

Output Size: 4800 samples by 5280 lines
```

**used by:**

- `create_dem` (p. 62)

**see also:**

- `escher` (p. 80), `zeroify` (p. 127)

**Notes:**
**projprm**

**Description:**

*projprm* is a program that creates a projection parameter table.

**Usage:**

```bash
projprm [ -log <file> ] projection prjkey prjunits <outfile> [ options ]
```

**Input:**

- **projection** – GEOGRAPH
  - PLSTEROE (Polar Stereographic)
  - UTM (Universal Transverse Mercator)
  - LAMAZEQA (Lambert Azimuthal Equal Area)
  - ALBERS (Albers Equal Area)
- **prjkey** – user defined projection key

**Output:**

- **outfile** – projection parameter table (created or updated if file exists)

**Options:**

- **-log** – allows the output to be written to a log file
- **options** – each projection has unique options required

<table>
<thead>
<tr>
<th>PROJECTION</th>
<th>OPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>GEOGRAPH</td>
<td>No other options</td>
</tr>
<tr>
<td>optional:</td>
<td></td>
</tr>
<tr>
<td>-d datum</td>
<td>Reference datum. Sets the semimajor axis and</td>
</tr>
<tr>
<td></td>
<td>eccentricity squared for the particular</td>
</tr>
<tr>
<td></td>
<td>projection. Default is 0 (Clarke 1866), use</td>
</tr>
<tr>
<td></td>
<td>12 for WGS 84.</td>
</tr>
</tbody>
</table>

| PLSTEROE     | Projection units must be FEET or MET.       |
| -l longitude | Center Longitude. This is the longitude that |
|              | the map is centered upon. Must be in same   |
|              | units as determined by the -g option.       |
| -p stdparl   | Standard Parallel. The latitude where the   |
|              | scale will be true. This is also the latitude|
|              | where the projection surface is tangent to  |
|              | or intersects the surface of the Earth.     |
|              | Must be same units as determined by the -g  |
|              | option.                                     |
| optional:    |                                             |
| -g geounits  | Units which -l and -s are entered; can be   |
|              | DEG (degrees), MIN (minutes), SEC (seconds), |
|              | or DMS (Degrees Minutes Seconds +DDDDMMSSSS|
|              | .SS). Defaults to DEG.                      |
| -d datum     | Reference datum. Sets the semimajor axis    |
|              | and eccentricity squared for the particular |
|              | projection. Default is 0 (Clarke 1866), use  |
|              | 12 for WGS 84.                              |
| -s smajoraxs | The semimajor axis of the Earth's ellipsoid |
|              | in meters. Default is NULL (0).             |
| -q ecsqval   | Eccentricity Squared Value. This value is    |
|              | treated in three different ways.             |
1. If value is zero, the earth is assumed to be a sphere.
2. If the value is greater than 0 and less than or equal to 1, the value is the eccentricity squared of the ellipsoid.
3. If the value is greater than 1, it is assumed to be the semimajor axis of the ellipsoid. Defaults to 0.

-e falseeast False easting. Offset of central meridian entered in meters. Defaults to 0.

-n falsenorth False northing. Offset of the origin of the projection, entered in meters. Defaults to 0.

**UTM**
Projection units must be FEET or MET, and must be specified either with the zone number or by their latitude/longitude values.

-z zonenum Zone Number. This is the number of the UTM zone in which the input coordinates are contained.

-a latitude Latitude. The latitude of any point within the UTM zone.

-l longitude Longitude. The longitude of any point within the UTM zone.

Optional:
-g geounits Unit of measure in which -l and –a are specified. Defaults to DEG.

-d datum Reference datum. Sets the semimajor axis and eccentricity squared for the particular projection. Set to 0 (Clarke 1866) by default. You may also want to use 12 (WGS 84).

**LAMAZEQA**
Projection units must be FEET or MET.

-x latitude Center Latitude coordinate. This is the latitude that the projection is centered upon. Specify in same units as -g option.

-y longitude Center Longitude coordinate. This is the longitude that the projection is centered upon. Specify in same units as -g option.

-r radius Radius of reference sphere. If 0 is specified, 6370997 meters is used.

Optional:
-e falseeast False easting. Offset of central meridian entered in meters. Defaults to 0.

-n falsenorth False northing. Offset of the origin of the projection, entered in meters. Defaults to 0.

-g geounits Units which -x and -y are specified in. Can be DEG (degrees), MIN (minutes), SEC (seconds), or DMS (Degree Minutes Seconds +DDDMMSSS.SS). Defaults to DEG.
### ALBERS

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a stdpar1</td>
<td>First Standard Parallel Parameter. Latitude of the first standard parallel. Specify in same units as -g option.</td>
</tr>
<tr>
<td>-b stdpar2</td>
<td>Second Standard Parallel Parameter. Latitude of the second standard parallel. Specify in same units as -g option.</td>
</tr>
<tr>
<td>-c centmer</td>
<td>Central Meridian. Longitude of the central meridian of the projection. Specify in same units as -g option.</td>
</tr>
</tbody>
</table>

**Optional:**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-g geounits</td>
<td>Units which -x and -y are specified in. Can be DEG (degrees), MIN (minutes), SEC (seconds), or DMS (Degree Minutes Seconds +DDDMMMSSS.SS). Defaults to DEG.</td>
</tr>
<tr>
<td>-o origin</td>
<td>Latitude of projection origin. Defaults to 0.</td>
</tr>
<tr>
<td>-s smaxis</td>
<td>Semi-Major Axis of the ellipsoid. If none is specified Clarke 1866 in meters is used.</td>
</tr>
<tr>
<td>-e ecsqval</td>
<td>Eccentricity squared of the ellipsoid if less than zero. If none is specified, the ellipsoid is assumed to be a sphere. If greater than zero, the semi-major axis of the ellipsoid is used.</td>
</tr>
<tr>
<td>-d datum</td>
<td>Reference datum. Sets the semimajor axis and eccentricity squared for the particular projection. Set to 0 (Clarke 1866) by default. You may also want to use 12 (WGS 84).</td>
</tr>
</tbody>
</table>

### Program log:

```plaintext
projprm -log projprm.log utm utm6 utm.proj -z 6
Date: Sun Oct 28 21:44:48 2001
Program: projprm
  Projection=UTM
  Projection Key=utm6
```

**see also:** [geocoding (p. 88)](#)
propagate

Description:

*propagate* is a script that propagates state vectors. The integral part of the propagation is the artificial satellite analysis program, which is an ephemeris propagation program for orbiting planetary spacecrafts. It uses Cowell's method of special perturbation. It includes harmonics of up to 40x40 field, lunar-solar gravity, drag, and solar radiation pressure. It uses a Runge-Kutta 8th order method with step size control for numerical integration. The program assumes a planet mean equator of epoch system and ignores polar motion.

Usage: propagate <inVectfile> <outVectfile>

Input: inVectfile – input state vector file

Output: outVectfile – propagated output state vector file

Program log:

propagate inVectfile outVectfile

Notes:
**refine_base**

**Description:**

*refine_base* is a program that refines the baseline and checks the validity of the result.

**Usage:**

```
refine_base [ -k iter ] [ -log <file> ] [ -quiet ]
     <phase> <tie_points> <meta> <old_base> <new_base>
```

**Input:**

- **phase** – unwrapped phase
- **tie_points** – seed point file with the seed points
- **meta** – meta data file
- **old_base** – old baseline file

**Output:**

**new_base** – refined baseline file

**Options:**

- **-k iter** – flag to keep intermediate results
- **-log** – allows the output to be written to a log file
- **-quiet** – suppresses the output to the essential

**Program log:**

```
Command line: refine_base -quiet -log refine_base.log unwrap_nod.phase out_dem_seeds a.meta base.00 base.01
Date: Fri Oct 26 16:36:28 2001
Program: refine_base

Baseline:   Normal: 90.596897, delta: -150.175736
            Parallel: -30.835124, delta: 68.617179
            Temporal: -0.999944 days

New Baseline: Normal: 88.023209, delta: 154.946884
               Parallel: -19.008820, delta: -68.038956
```

**used by:**

- *create_dem* (p. 62)

**see also:**

- *deramp* (p. 69), *escher* (p. 80), *resolve* (p. 118)

**Notes:**
refine_offset

Description:

refine_offset is a program that refines the slant range and time offset in the metadata file in order to improve the geolocation of the final result. It uses a set of points with known geolocation. If the area covers Delta Junction the number of the respective corner reflectors can be used to identify the geolocation. Otherwise the geolocation of the point has to be given in latitude, longitude and elevation (meters).

Usage: refine_offset <metadata> <ddr_file> [ DJ# | locSpec ] y x

Input: metadata – image metadata
       ddr_file – data descriptor file

Output: none

Options: DJ# – corner reflector number (Delta Junction)
          locSpec – geolocation (latitude/longitude/elevation in meters)
          x – sample position of point (pixels)
          y – line position of point (pixels)

Program log:

refine_offset a.meta sar_byte.img 64.859485/-147.849574/215 6067.4 679.6

Finding lat=64.859485, lon=-147.849574, elev=215.000000
Positional Error: 1326.899360 m north, 1169.319591 m east
Offsets: Time=0.280057 s; Slant Range=243.795514 m

…

Offsets: Time=0.242303 s; Slant Range=259.207822 m
Positional Error: -0.000262 m north, 0.000051 m east

see also: geocode (p. 88)

Notes:
remap

Description:

remap is a program that performs the remapping and resampling of images. Remapping an image implies that the location is changed, while resampling changes the actual pixel value in the output image. The program is able to handle a large variety of options using different mapping functions, sampling methods, data types and handling different images sizes.

Usage: remap

map: [ -rotate <deg> ] [ -scale sX sY ] [ -translate tX tY ]
[ -matrix <matrixFile> ] [ -ppf <ppfFile> ] [ -deltas <deltaFile> ]
[ -quadratic <quadFile> ] [ -warp <warpImages> ]
sample: [ -nearest ] [ -bilinear ] [ -sinc ] [ -kernel sizeX sizeY ]
[ -fileKernel <kernelFile> ]
data type: [ -background fill ]
[ -byte ] [ -map min max ] [ -char ] [ -short ] [ -int ] [ -float ] [ -double ]
size: [ -width width ] [ -height height ] [ -sameSize ]
[ -asDDR <ddr> ]
[ -log ]
<infile> <outfile>

Input: infile – input file
Output: outfile – output file

Options: -rotate – clockwise rotation (in degrees)
    -scale – scaling factor
    -translate – translation parameters (upper left corner of the image is origin of coordinate system)
    -matrix – image transformation matrix file
    -ppf – ppf coefficient file
    -deltas – offset parameter
    -quadratic – quad file (least-squares fit)
    -warp – horizontal and vertical offset grid files
    -nearest – nearest neighbor interpolation
    -bilinear – bilinear interpolation (default value)
    -sinc – sinc kernel for interpolation
    -kernel – uniform kernel of variable size → low pass filtering
    -fileKernel – predefined kernel file
    -background – background fill
    -byte – byte map
    -char – character byte file
    -short – short real number file
    -float – floating point file (fastest data type to be calculated)
-double – double precision file
-width – width of the output image
-height – height of the output image
-sameSize – flag for same size of input and output image
-asDDR – ddr file from which size and projection information is read
-log – allows the output to be written to a log file

Program log:

Command line: remap -matrix dem_plane -translate 0 0 -width 5200 -height 5280 -bilin -float -log remap.log
alaska.img dem_big.dem
Date: Sat Sep 8 13:14:43 2001
Program: remap

Input dimensions: 32673x40035 (LxS).
Output dimensions: 5280x5200 (LxS).
Output Pixel type/size: Floating point pixels (32-bit IEEE).
Sampling Type: Bilinear.

used by: create_dem (p. 62)

see also: fit_plane (p. 85), fit_quadratic (p. 86), fit_warp (p. 87), resample (p. 116)

Notes:
resample

Description:

resample is a program that resamples the input file to a desired output pixel resolution. The program determines the size of the filtering kernel from the desired output pixel size and subsamples the image. It only works with byte images.

Usage: resample <infile> <outfile> <pixsize>

Input: infile – input file
pixsize – pixel size of output image

Output: outfile – output file

Program log:

resample r1_39153.img r1_39153_low.img 100

Resample: Performing filtering and subsampling..

Input image is r1_39153.img
Output image is r1_39153_low.img
Output pixel size is 100.000000 meters
Scale Factor : 0.500000
Kernel Size : 3
base-- 0.500000 rate-- 2.000000

Writing raw byte image file 5500 samples by 5500 lines

Processing Output Line 0
Processing Output Line 40
...
Processing Output Line 5440
Processing Output Line 5480
Total wall clock time = 15.910000 seconds.

see also: remap (p. 114)

Notes:
reskew_dem

Description:

reskew_dem is a program that remap a ground range elevation model into a slant range elevation model and creates a slant range simulated amplitude. The height in slant range $h_{SR}$ is calculated by

$$h_{SR}(x) = h_{GR}(x) + \frac{h_{GR}(x) - h_{GR}(x-1)}{pos_{SR}(x) - pos_{SR}(x-1)}$$  (5-16)

with the height in ground range $h_{GR}$, the pixel positions in ground range $pos_{GR}$ and in slant range $pos_{SR}$.

Usage:    reskew_dem <inGR_DEM> <meta> <outSR_DEM> <outSR_simAmp>

Input:    inGR_DEM – ground range elevation model
          meta – metadata SAR image

Output:   outSR_DEM – slant range elevation model
          outSR_simAmp – simulated slant range amplitude

Options:  -log – allows the output to be written to a log file

Program log:

Command line: reskew_dem dem_big.dem a.meta dem_slant.ht dem_sim.amp -log reskew_dem.log
Date: Sat Sep 8 13:19:33 2001
Program: reskew_dem

Converted 5280 lines from ground range to slant range.

used by:  create_dem (p. 62)

see also: deskew_dem (p. 71)
**resolve**

**Description:**

`resolve` is a program that estimates the initial offset between master and slave image. The first estimate for the offset is derived from the state vectors. This offset is then refined using a correlation match in the frequency domain (`fftMatch`). A first estimate of the baseline is stored as well as a parameter file used for the fine coregistration (`fico`).

**Usage:**  
```bash  
resolve [ -log <file> ] [ -quiet ] <image 1> <image 2> <basefile> <ctrlfile>  
```

**Input:**  
`image 1`, `image 2` – master and slave image in cpx format

**Output:**  
`basefile` – baseline information file  
`ctrlfile` – parameter file used as input for `fico`  
`correlation file` – correlation image “reg_cor” stored to check the correlation quality

**Options:**  
- `-log file` – allows the output to be written to a log file  
- `-quiet` – suppresses the output to the essential

**Program log:**

Command line: `resolve -log resolve.log -quiet reg/a_p1 reg/b_p1 base.00 reg/ctrl1`

Date: Sat Sep 8 13:04:48 2001

Program: `resolve`

Baseline:  
```text  
Bn = 67.828500, dBn = 88.450432, Bp = -25.023166, dBp = -38.157228, Btemp = -0.999629  
```

Offset slave image:  
```text  
dx = 2.986450, dy = -42.292938  
```

Certainty: 56.133865%

Complex image offset is -211 rows, 2 columns

**used by:**  
`create_dem()`

**see also:**  
`aisp` (p. 41), `fftMatch` (p. 82), `fico` (p. 83)

**Notes:**
**sarin**

**Description:**

`sarin` is a program that ingests a CEOS format detected SAR image into the internal generic binary format. Single look complex (SLC) CEOS format data are ingested using `trim_slc`.

**Usage:**

```
sarin <inSARfile> <outfile>
```

**Input:**

`inSARfile` – input detected SAR image in CEOS format

**Output:**

`outfile` – output image in generic binary format

**Program log:**

```
sarin R139153153G3S006.D r1_39153

  Data set processed by ASF
  ALPHA123 new x-vector: -0.107,0.104,0.989
  alpha1, alpha2, alpha3  315.704410 -98.577445 -31.168281

 Data set processed by ASF
  ALPHA123 new x-vector: -0.107,0.104,0.989
  alpha1, alpha2, alpha3  315.704410 -98.577445 -31.168281
  Input and output nl=11000, ns=11000
  Total wall clock time = 8.010000 seconds.
```

**see also:** `trim_slc` (p. 126), `sarout` (p. 120), `ceos2raw` (p. 52)

**Notes:**
sarout

Description:

sarout is a program that converts images that are formatted in the internal generic binary format into ASF CEOS formatted products. The inputs can be of type .raw, .cpx, or .img to create output products of type CCSD, SLC, or detected images, respectively.

Usage: sarout [ -L inCeos ] < m | c | x | d > <inMeta> <inImg> <outCeos>

Input:  
inMeta – input metadata file
inImg – input image
m – convert .raw into CCSD metadata (.L) file
c – convert .raw into CCSD
x – convert .cpx into COMPLEX
d – convert .img to DETECTED IMAGE

Output: outCeos – output image in CEOS format

Options: -L inCeos – Modified CEOS file with all original information.

Program log:

sarout x r1_33408 r1_33408 R133408215X001

==================================
ASF PRODUCT GENERATION TOOL
==================================

Creating ASF COMPLEX product
There is a problem with the .ant or .noise file

Meta file added to leader records
Histogram Structure Initialized
...Converting Line 0
...Converting Line 500
...
...Converting Line 33500
...Converting Line 34000
Finished with leader file descriptor record.
Finished with data set summary record.
Finished with platform position data record.
Finished with the attitude data record.
Finished with the radiometric data record.
Finished with data quality summary record.
Finished with signal data histogram record.
Finished with processed data histogram record.
Finished with range spectra record.
Finished with facility related data record.
Total wall clock time = 163.729996 seconds.
see also: sarin (p. 119), trim_slc (p. 126), ceos2raw (p. 52)

Notes:
sr2gr

Description:

sr2gr is a program that converts an image from slant range into ground range projection. The algorithm calculates the ground range to the first pixel in the image from the spacecraft ephemeris, earth ellipsoid, and slant range to first pixel given in the image's metadata. It then uses the slant range spacing interval to determine appropriate ground range positions. The remapping is performed using bi-linear interpolation.

Usage:  
sr2gr <infile> <inmeta> <outfile> <outmeta> pixsize

Input: 
infile – input image in slant range projection
inmeta – input image metadata
pixsize – pixel size of the output image

Output:  
outfile – output image in ground range projection
outmeta – output image metadata

Program log:

sr2gr r1_33408_sr.amp r1_33408_sr.meta r1_33408_gr.amp r1_33408_gr.meta 12.5

Output Pixels: 12.500000 (range) 12.500000 (azimuth)
Input image is r1_33408_sr.amp
Input lines, samples: 34279 7408
Output image is r1_33408_gr.amp
Output lines, samples: 13492 3983

...writing line 0
...writing line 200
...
...writing line 13200
...writing line 13400

see also:  deskew (p. 70)

Notes:
stats

Description:

stats is a program that calculates statistics for a SAR data file. It writes the statistics with a histogram out to a .stat file and inserts a stats block in the .meta file. The statistics can also be calculated for a subset of the image.

Usage: stats [ -mask <value> ] [ -log <logFile> ] [ -quiet ] [ -overmeta ] [ -nometa ] [ -overstat ] [ -nostat ] [ -startline <line> ] [ -startsamp <sample> ] [ -width <width> ] [ -height <height> ] <sar_name>

Input: sar_name – input file for which statistics needs to be calculated

Options: -mask – ignores mask value while calculating statistics
 -log – copies terminal output to <logFile>
 -quiet – supresses terminal output
 -overmeta – forces overwrite of existing .meta file
 -nometa – does not write a .meta file
 -overstat – forces overwrite of existing .stat file
 -nostat – does not write a .stat file
 -startline – starts counting at image line <line> (implies -nometa)
 -startsamp – starts counting at image sample <sample> (implies -nometa)
 -height – only processes <height> lines (implies -nometa)
 -width – only processes <width> samples per line (implies -nometa)

Program log:

stats -log stats.log r1_31901.img

Date: Fri Oct 3 15:39:46 2003
Program: stats

Statistics found:
Minimum = 0
Maximum = 255
Mean = 27.885725687
Root mean squared error = 0.002535296453
Standard deviation = 0.0025352964635

Data fit to [0..255] using equation: byte = 1 * sample + 0

Histogram:
0- 7: 35515125 0 0 0 0 0 16 147 739
8-15: 3070 9308 22164 44686 79071 125531 187433 262746
16-23: 354339 464916 595483 743290 909692 1094517 1287967 1494602
24-31: 1704229 1911857 2112152 2297119 2462974 2598154 2711367 2798024
32-39: 2851692 2879119 2878136 2850537 2805043 2743009 2633488 2574927
40-47: 2479080 2372696 2262662 2153708 2039898 1928401 1817100 1707062
48-55: 1599263 1495404 1393753 1295811 1204431 1116235 1031921 955677
56-63: 878730 810157 742523 679464 624271 570009 520973 477214
| 64-71 | 434435 396350 360129 327341 297662 269558 245953 222841 |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 72-79 | 202046 183265 166144 150966 137158 123542 112290 101518 |
| 80-87 | 92525 83237 75666 68312 61856 56478 50672 46208 |
| 88-95 | 41987 37940 34815 31139 28840 25912 23973 21264 |
| 96-103| 19709 17897 16507 14923 13613 12481 11593 10614 |
| 104-111| 9467 8960 8191 7627 7041 6368 5997 5418 |
| 112-119| 5131 4889 4482 4147 3906 3583 3374 3047 |
| 120-127| 2913 2754 2620 2481 2311 2216 2058 1984 |
| 128-135| 1866 1799 1656 1600 1469 1485 1405 1386 |
| 136-143| 1305 1223 1206 1074 1082 1050 1012 1030 |
| 144-151| 966 895 896 848 831 824 761 766 |
| 152-159| 721 691 681 636 656 608 588 582 |
| 160-167| 531 578 601 537 496 555 562 513 |
| 168-175| 475 460 490 455 421 459 407 456 |
| 176-183| 394 407 430 343 370 391 320 359 |
| 184-191| 360 329 315 347 320 310 321 325 |
| 192-199| 307 294 307 269 264 295 288 281 |
| 200-207| 261 241 224 257 236 247 263 223 |
| 208-215| 231 235 252 185 222 217 221 202 |
| 216-223| 205 202 181 201 167 206 176 177 |
| 224-231| 180 207 176 191 150 171 153 134 |
| 232-239| 130 176 161 155 141 144 127 154 |
| 240-247| 140 116 141 142 154 119 128 127 |
| 248-255| 142 118 148 124 110 108 115 7188 |

Statistics taken on image file 'r1_31901.img'
Statistics written to the stats block in r1_31901.meta
Statistics plus histogram written to r1_31901.stat

Total wall clock time = 32.169998 seconds.

see also: diff_las (p. 72), convert2byte (p. 60)

Notes:
trim

Description:

trim is a program that changes the size of any LAS image.

Usage: trim [ -log <file> ] <infile> <outfile> new_top_line new_left_sample [ trim_new_height trim_new_width ]

Input:  
infile – input image
new_top_line – starting line of the output image
new_left_sample – starting sample of the output image

Output:  
outfile – output LAS image

Options:  
-log – allows the output to be written to a log file
trim_new_height – height of the output image
trim_new_width – width of the output image

Program log:

Command line: trim -log trim.log dem_simbyte.img dem_trimsim.img 0 0 5280 4800  
Date: Sat Sep 8 13:19:56 2001  
Program: trim

Wrote 5280 lines of data

used by:  
create_dem (p. 62)

see also:  
trim_slc (p. 126)

Notes:
**trim_slc**

**Description:**

*trim_slc* is a program that converts a single look complex image into a LAS float complex image and allows trimming the output image in size.

**Usage:**

```plaintext
trim_slc <inSLCfile> <outCPXfile> new_top_line new_left_sample
    [ trim_new_height  trim_new_width ]
```

**Input:**

- `inSLCfile` – single look complex image
- `new_top_line` – starting line of the output image
- `new_left_sample` – starting sample of the output image

**Output:**

- `outCPXfile` – complex float LAS image

**Options:**

- `trim_new_height` – height of the output image
- `trim_new_width` – width of the output image

**Program log:**

```
trim_slc frame.slc frame 0 0 2500 1000

trim_slc: input SLC: 'frame.slc'. output Float-Complex: 'frame.cpx'.
    Input image size: Lines=24099, Samples=6603
    Output image size: Lines=2500, Samples=1000
    Writing output image line 0...
    Writing output image line 1000...
    Writing output image line 2000...
trim_slc completed successfully!
```

**used by:**

- create_dem (p. 62)

**see also:**

- resolve (p. 118), trim (p. 125)

**Notes:**
zeroify

Description:

zeroify is a program that creates an output image which is identical to the input image, but zero where the test image is zero.
This program is used to set the phase values in the filtered phase image to zero that are zero in the phase simulated from the reference DEM.

Usage:

zeroify <input image> <test image> <output image>

Input:

input image – filtered phase image
test image – phase image simulated from the DEM

Output:

output image – corrected filtered phase image

used by:

create_dem (p. 62)

see also:

escher (p. 80), phase_filter (p. 107)

Notes:
This page is intentionally blank
GLOSSARY

Across-track direction. see Range direction.

Across-track interferometry. The across-track method of interferometry is presently only employed on airborne systems as it requires two SAR antenna systems to be mounted simultaneously on the platform. The position of the two antennas mounted on an aircraft in this configuration is perpendicular to the flight direction. A problem in this geometry is the fact that the distinction between errors caused by the aircraft roll and the influence of the terrain slope is not possible.

Along-track direction. see Azimuth direction.

Along-track interferometry. The along-track method of interferometry is only applicable at present to airborne SAR systems, as both along-track and across-track methods require two antenna systems on the same platform. The yaw and pitch cause baseline components in y- and z-direction which produce additional phase differences. Before absolute velocity measurements are possible, a calibration of the phase difference is necessary.

Amplitude. The amplitude is a measure of the strength of the signal. In case of a complex signal it includes both the magnitude and the phase.

Antenna. The antenna is a device to radiate electromagnetic energy on transmission by a radar, and to collect such energy during reception. An antenna pattern is designed with spatial directivity, which concentrates the energy into a beam in both the vertical and the horizontal directions.

Antenna footprint. The antenna footprint covers the area on the ground hit by one transmitted pulse and from which the backscattered signal will be received.

Ascending orbit. The ascending orbit is a satellite orbit crossing south-north with a radar sensor looking to the east. The image is rotated towards west in this case.

Azimuth direction. The azimuth direction, often referred to as along-track direction, is defined as the flight direction of the satellite or airborne system.

Backscatter coefficient. The backscatter coefficient is defined as the radar cross section divided by the minimum resolved area illuminated on the target.

Bandwidth. The bandwidth is a measure of the span of frequencies available in a signal or of the frequency limiting stages in the system. It is a fundamental parameter of any imaging system and determines the resolution available.
**Baseline.** The baseline is defined as the separation between the two antenna positions either mounted on an aircraft or realized by two repeating satellite orbits. The length of the baseline is taken as a measure for the \( \rightarrow \text{coherence} \) expected for SAR interferometric data. A precise knowledge of the length and the position of the baseline is a prerequisite for a good performance of the data processing. See also \( \rightarrow \text{critical baseline} \).

**Baseline components.** The baseline geometry can be represented in three different ways. The parallel flight paths are separated by the baseline with an orientation angle \( \alpha \). This baseline can also be divided into the horizontal and the vertical component as well as into the parallel and the perpendicular component of the baseline.

**Baseline decorrelation.** Since the two antennas are separated in the \( \rightarrow \text{across-track direction} \), slightly different interference results are measured, even when the pixel derived from each antenna covers the same area. This results in a loss of \( \rightarrow \text{coherence} \) between the two antenna outputs, referred to as baseline decorrelation.

**Coherence.** The coherence is a measure for the correlation of the phase information of corresponding signals. It ranges from 0 to 1.

**Critical baseline.** The critical baseline is reached when the \( \rightarrow \text{baseline} \) becomes larger than half of the reflected beam width. In that case, the radar echoes have lost the \( \rightarrow \text{coherence} \).

**Decorrelation.** The decorrelation is a measure of the degree of phase \( \rightarrow \text{coherence} \) in the complex radar returns from a target of interest acquired at different times. It has three uncorrelated components. The thermal component refers to the \( \rightarrow \text{thermal noise} \) introduced by the system. The spatial aspect is covered by effects related to the \( \rightarrow \text{baseline decorrelation} \). Finally, there is a temporal component describing the \( \rightarrow \text{temporal decorrelation} \).

**Descending orbit.** The descending orbit is a satellite orbit crossing north-south with a radar sensor looking to the west. The image is rotated towards east in this case.

**Differential radar interferometry.** The differential use of SAR interferometry is adapted for the measurement of small-scale movements in the vertical direction, e.g. for change detection. This technique provides relative measures of the order of a few centimeters or even less. The displacement measured by differential interferometry is not vertical, but along the viewing direction.

**Doppler centroid.** The Doppler centroid is the center Doppler frequency or zero of the Doppler spectrum as the radar beam sweeps past the target. It is found by correlating power spectra with some predefined weight functions. Since the Doppler centroid generally varies over range, estimation is performed at several range positions. The Doppler centroid estimate is required to maximize \( \rightarrow \text{signal-to-noise-ratio} \) and minimize azimuth ambiguities by matching the pass band of the azimuth compression filter to the spectral energy distribution of the SAR signal.
**Far range.** The far range refers to the portion of a radar image farthest from the flight path.

**Footprint.** see *Antenna footprint.*

**Foreshortening.** The time difference of two signals backscattered at the bottom and the top of a steep slope is shorter than in a flat area. Therefore, the distance between two points is mapped shorter in the image. This geometric effect called foreshortening compresses the backscattered signal energy, i.e. the affected area on the image appears brighter.

**Frame.** An → *orbit* is split in several standard scenes called frames whose size and location depend on the satellite. The ERS satellite orbit, for example, consists of 7200 nodes and 400 of them are used to identify the SAR frames through the node closest to the frames’ center. The standard frame size of an ERS scene is 100 → 100 km.

**Frequency.** The frequency is the rate of oscillation of a wave. In the microwave range, the frequencies are on the order of 1 GHz to 100 GHz.

**Fringe.** A fringe represents the whole range of the phase in an interferogram from 0 to 2π in a full color cycle.

**Ground control point.** The ground control point (GCP) refers to a physical point on the ground whose ground positions are known with respect to some horizontal coordinate system.

**Ground range.** The ground range is the distance from the → *nadir* to a given object.

**Incidence angle.** The incidence angle is the angle between the radar beam center and the normal to the local topography.

**Inclination.** The inclination is the angle of the plane of a satellite orbit with respect to the equatorial plane of the earth.

**Intensity.** The intensity is a measure of the energy reflected or emitted by an object.

**Interferogram.** The interferogram is defined as the product of the complex SAR values of the one image with the complex conjugate of the another image; i.e. the corresponding amplitudes have to be averaged and the corresponding phases have to be differenced at each point in the image. It contains the → *amplitude* as well as the → *phase* information of a SAR image pair.

**Interferometric processing.** The interferometric processing generally consists of the registration of the complex images, the formation of the → *interferogram*, the → *phase unwrapping* and the reconstruction of the digital elevation model.
Layover. The layover effect represents the extreme case of \( \rightarrow \text{foreshortening} \). The signal backscattered from the top of the mountain is received earlier than the signal from the bottom, i.e. the foreslope is reversed. The pixel information from various objects is superimposed which leads to a brighter appearance on the image.

**Line spacing.** The line spacing represents how much area each pixel covers in \( \rightarrow \text{azimuth direction} \).

**Look angle.** The look angle is defined as the angle between the direction the antenna is pointing when transmitting and receiving from a particular cell and the \( \rightarrow \text{nadir} \).

**Looks.** The frequency varies linearly with time so that the spectrum may be obtained by dividing the coherent integration time into sequential subintervals and computing the energy returned during each interval. These data segments represent the application of the pulse compression processing over only a part of the synthetic aperture and are referred to as looks.

**Magnitude.** The magnitude is the \( \rightarrow \text{amplitude} \) of the wave irrespective of the \( \rightarrow \text{phase} \).

**Nadir.** The nadir is defined as the direction towards the center of the Earth.

**Near range.** The near range refers to the portion of a radar image closest to the flight path.

**Noise.** Noise is defined as random or regular interfering effects in the data which degrade its information-bearing quality. See also \( \text{speckle} \).

**Orbit.** The orbit is the curved path, usually elliptical, described by a satellite about the earth.

**Phase.** The phase is the angle of a complex number.

**Phase unwrapping.** The resultant \( \rightarrow \text{phase} \) of the \( \rightarrow \text{interferogram} \), which is directly related to the topography, is only measured modulo \( 2\pi \). To calculate the elevation of each point it is necessary to add the correct integer number of phase cycles to each phase measurement. The problem of solving this \( 2\pi \) ambiguity is called phase unwrapping.

**Phasor.** The phasor is the vector representing the resultant of all scattering objects within the \( \rightarrow \text{resolution cell} \).

**Pixel.** Acronym for picture element. The pixels form the image in a regular grid and visualize the data. The size of the pixel can exceed the size of the \( \rightarrow \text{resolution cell} \) in order to obtain square pixels or to reduce the \( \rightarrow \text{noise} \).

**Pixel spacing.** The pixel spacing represents how much area each pixel covers in \( \rightarrow \text{range direction} \).
Posting. The posting is defined as the distance between the centers of the resolution cell and can be unequal to the size of the resolution cell.

Pulse repetition frequency. The pulse repetition frequency (PRF) determines the number of pulses transmitted per second.

Radar cross section. The radar cross section is the area of a fictitious, perfect reflector of electromagnetic waves that would reflect the same amount of energy back to the radar as the actual target.

Radiometric resolution. Radiometric resolution is a measure of how different two uniform (speckled) regions of different backscatter levels have to be in order to distinguish between them.

Range. The range is the distance from the radar to the target perpendicular the flight direction. It can be either represented as slant range or as ground range.

Range direction. The range direction, often referred to as across-track direction, is defined as the direction perpendicular to the flight direction of the satellite or airborne system.

Repeat cycle. The repeat cycle is the time between one pass over a fixed location on earth and the subsequent pass over the same location. It is identified by the number of orbits between the two repeated passes. More practically, but less accurately it is identified by the number of days between repeat coverages.

Repeat-pass interferometry. Repeat-pass interferometry requires only one antenna and hence is the method most suited to spaceborne SAR sensors. This is also because for this method precise location of the flight path is required, and satellites typically have much more precise and stable orbital paths in the absence of the atmosphere, than do aircraft. The satellite has to pass nearly the same orbit to cover an area twice with a slightly different viewing geometry.

Resolution. The resolution is a measure for the minimum (spatial) separation between two measurements in order for a sensor to be able to discriminate between them. The range resolution depends on the pulse length whereas the azimuth resolution is determined by the angular beam width of the illuminated terrain strip. Spectral and radiometric resolutions refer to the resolving power of a system in wavelength and energy, respectively.

Resolution cell. The resolution cell is an element of a SAR image representing the physical properties. Besides the physical properties, the size of the resolution cell is also determined by the processing from raw data to single look complex (SLC) data. See also pixel, posting.
**SAR processing.** SAR processing, often referred to as SAR correlation, matched filtering or azimuth compression, consists of image focusing through matched filter integration, detection, and multi-look summation. The output files of a SAR processor usually are presented with unity aspect ratio (so that range and azimuth image scales are the same). Images may either be in → *slant range* or → *ground range* projection. Both of these spatial adjustments require resampling of the image file.

**Shadow.** The shadow effect in radar imagery is different from optical imagery. In the case of → *radar*, no information is received from the backslope which appear as black regions on the image. The length of the shadow depends on its position in range direction. Therefore, the shadow in → *far range* are longer than in → *near range*.

**Signal-to-noise-ratio.** The signal-to-noise-ratio (SNR) is the ratio of the level of the information-bearing signal to the level of the noise power. The maximum SNR of a device is called the “dynamic range”.

**Single look complex.** The single look complex (SLC) data contain signal information in a complex number, i.e. the → *intensity* and the → *phase* information is stored in a real and an imaginary part.

**Slant range.** The slant range represents the distance from radar sensor to the target.

**Speckle.** Speckle is a scattering phenomenon which arises because the resolution of the sensor is not sufficient to resolve individual scatterers. Physically speaking speckle is not → *noise*, since the same imaging configuration leads to the identical speckle pattern. Speckle can be reduced by multi-look processing and spatial averaging.

**Swath.** The swath is the strip of ground swept over by the moving radar beam.

**Synthetic aperture radar.** In a synthetic aperture radar (SAR) the azimuth resolution is achieved through computer operations on a set of (coherently recorded) signals such that the processor is able to function such as a large → *antenna* aperture in computer memory, thus realizing azimuth improvement in proportion to aperture size.

**Temporal decorrelation.** The temporal decorrelation is caused by physical changes in the surface over the time period between observations.

**Track.** Each → *repeat cycle* has an internal number of → *orbits* called tracks which have the property of always covering the same area. For the ERS satellite the track numbers follow the orbit numbers.

**Wavelength.** The wavelength is defined as the velocity divided by the → *frequency* of a wave.
## INDEX

<p>| .000 file | 149 | base.00 file | 172 |
| .amp file | 149 | Baseline | 7 |
| .chop file | 149–50 | Critical length | 7 |
| .cpx file | 150 | Definition | 7 |
| .D file | 150 | Estimation | 7 |
| .dat file | 150 | Length | 7 |
| .ddr file | 151–52 | Possible representations | 7 |
| .dem file | 152 | Baseline refinement | 22 |
| .fnt file | 152 | c2p | 48 |
| .ht file | 152 | calc_deltas | 49 |
| .img file | 153 | calibrate | 50–51 |
| .in file | 153 | CEOS frames | 17 |
| .L file | 153 | ceos2raw | 52–53 |
| .ldr file | 154 | Change detection | 11 |
| .log file | 154–58 | coh | 54–55 |
| .meta file | 160 | Coherence | 7 |
| .nul file | 162 | Forestry | 12 |
| .par file | 162–67 | Interferometric product | 9 |
| .phase file | 168 | Multibaseline technique | 8 |
| .pi file | 168 | Multifrequency approach | 11 |
| .proj file | 168–69 | Ocean currents | 11 |
| .raw file | 169 | Complex radar signal | 5 |
| .replica file | 169 | concat | 56 |
| .stat file | 170–71 | concat dem | 57 |
| .ti file | 171 | concat2byte | 58–59 |
| .trl file | 171 | convert2byte | 60–61 |
| .vol file | 172 | Coregistration | 19 |
| accum_offset | 40 | aisp | 41–42 |
| Across-track interferometry | 8 | calc_deltas | 49 |
| Airborne interferometry | 8 | ceos2raw | 52–53 |
| Across-track interferometry | 8 | fftMatch | 82 |
| Along-track interferometry | 8 | fico | 83 |
| First practical study | 3 | fit_line | 84 |
| Ocean currents | 11 | fit_plane | 85 |
| aisp | 41–42 | Offset | 19 |
| Albers equal area projection | 22 | remap | 114–15 |
| Along-track interferometry | 8–9 | resolve | 118 |
| Amplitude | 5 | interp | 19 |
| Applications | | | |
| Digital elevation modelling | 11 | l2raw_flywheel | 96–97 |
| Forestry | 12 | Offset | 19 |
| Ocean currents | 11 | remap | 114–15 |
| Polar research | 11 | resolve | 118 |
| Potential applications | 4 | Coseismic displacements | 12 |
| Seismic events | 11 | create_dem | 62 |
| Surface movements | 11–12 | create_dem_grid | 63 |
| Topographic mapping | 3, 4, 10 | Critical baseline | 7 |
| Volcanic hazards | 12 | Crossed orbits | 4 |
| Ascending orbit | 11 | ctrl file | 172–73 |
| Atmospheric effects | 4, 8, 11 | Data acquisition | 8 |
| auto_sar_mosaic | 43–44 | Data format | 17 |
| auto_swath_mosaic | 45–46 | Data handling | |
| avedop file | 172 | dspddr | 74–75 |
| avg_in_dop | 47 | trim_slc | 126 |
| Backscatter behaviour | 5 | Bandwidth | 5 |
| Bandwidth | 5 | Look angle | 5 |</p>
<table>
<thead>
<tr>
<th>Data quality (continued)</th>
<th>File formats (continued)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial resolution</td>
<td>.fnt 152</td>
</tr>
<tr>
<td>System parameters</td>
<td>.hm 153</td>
</tr>
<tr>
<td>Wavelength</td>
<td>.in 153</td>
</tr>
<tr>
<td>Data sets</td>
<td>.input vector file</td>
</tr>
<tr>
<td>CEOS frames</td>
<td>.L 153</td>
</tr>
<tr>
<td>Data format</td>
<td>.ldr 154</td>
</tr>
<tr>
<td>ERS-1 data</td>
<td>.log 154</td>
</tr>
<tr>
<td>ERS-2 data</td>
<td>.meta 160</td>
</tr>
<tr>
<td>Level zero data</td>
<td>.nul 162</td>
</tr>
<tr>
<td>SEASAT data</td>
<td>.par 162</td>
</tr>
<tr>
<td>SIR-B data</td>
<td>.phase 168</td>
</tr>
<tr>
<td>TOPSAR data</td>
<td>ddr2meta 64</td>
</tr>
<tr>
<td>Baseline decorrelation</td>
<td>168–69</td>
</tr>
<tr>
<td>Temporal decorrelation</td>
<td>.proj</td>
</tr>
<tr>
<td>Deformations</td>
<td>.raw</td>
</tr>
<tr>
<td>deltas file</td>
<td>.replica 169</td>
</tr>
<tr>
<td>DEM generation</td>
<td>.stat 170–71</td>
</tr>
<tr>
<td>Error map</td>
<td>176</td>
</tr>
<tr>
<td>Geocoding</td>
<td>.vol</td>
</tr>
<tr>
<td>Map projection</td>
<td>.avedop 172</td>
</tr>
<tr>
<td>Slant range to ground range</td>
<td>.base.00 172</td>
</tr>
<tr>
<td>dem_corr</td>
<td>.ctrl172–73</td>
</tr>
<tr>
<td>dem_grid</td>
<td>deltas 173</td>
</tr>
<tr>
<td>dem2phase</td>
<td>dem_corr 173</td>
</tr>
<tr>
<td>dem2seeds</td>
<td>dem_grid 173–74</td>
</tr>
<tr>
<td>deramp</td>
<td>176</td>
</tr>
<tr>
<td>Descending orbit</td>
<td>.line</td>
</tr>
<tr>
<td>deskew</td>
<td>.matchFile 176</td>
</tr>
<tr>
<td>deskew_dem</td>
<td>.matrixFile 176</td>
</tr>
<tr>
<td>diff_las</td>
<td>.out_dem_seeds 177</td>
</tr>
<tr>
<td>Differential interferogram</td>
<td>177</td>
</tr>
<tr>
<td>Differential interferometry</td>
<td>84</td>
</tr>
<tr>
<td>Digital elevation model</td>
<td>.fit_line</td>
</tr>
<tr>
<td>Applications</td>
<td>.fit_plane 85</td>
</tr>
<tr>
<td>Forestry</td>
<td>.fit_quad 86</td>
</tr>
<tr>
<td>Fringe</td>
<td>.fit_warp 87</td>
</tr>
<tr>
<td>Doppler</td>
<td>.Forestry 12</td>
</tr>
<tr>
<td>dsppdr</td>
<td>.Fringe 20</td>
</tr>
<tr>
<td>elev</td>
<td>.geocode 88–89</td>
</tr>
<tr>
<td>eleverr</td>
<td>.Geocoded products 10</td>
</tr>
<tr>
<td>Error map</td>
<td>.Geocoding 22</td>
</tr>
<tr>
<td>ERS-1 data</td>
<td>.projprm 108–10</td>
</tr>
<tr>
<td>Forestry</td>
<td>.Global positioning system</td>
</tr>
<tr>
<td>Geometry</td>
<td>.General viewing geometry</td>
</tr>
<tr>
<td>Geometry</td>
<td>.8</td>
</tr>
<tr>
<td>Global positioning system</td>
<td>.IGRAM 9</td>
</tr>
<tr>
<td>Interferometric phase</td>
<td>.Inertial navigation system</td>
</tr>
<tr>
<td>Input vector file</td>
<td>.Inertial navigation system</td>
</tr>
<tr>
<td>Interferogram</td>
<td>.Airborne interferometry</td>
</tr>
<tr>
<td>Interferogram / Coherence generation</td>
<td>.19–21</td>
</tr>
<tr>
<td>K</td>
<td>.coh 54–55</td>
</tr>
<tr>
<td>Fringe</td>
<td>.12</td>
</tr>
<tr>
<td>Topographic mapping</td>
<td>.6</td>
</tr>
<tr>
<td>Coherence</td>
<td>.9</td>
</tr>
<tr>
<td>Interferometric products (continued)</td>
<td>resample</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Differential interferogram</td>
<td>Resampling</td>
</tr>
<tr>
<td>Digital elevation model</td>
<td>reskew_dem</td>
</tr>
<tr>
<td>Geocoded products</td>
<td>resolve</td>
</tr>
<tr>
<td>Interferogram</td>
<td>SAR interferometry</td>
</tr>
<tr>
<td>Simulated interferogram</td>
<td>SAR processing</td>
</tr>
<tr>
<td>Interpolation</td>
<td>Doppler</td>
</tr>
<tr>
<td>Lambert azimuthal equal area projection</td>
<td>22</td>
</tr>
<tr>
<td>Land subsidence</td>
<td>sarin</td>
</tr>
<tr>
<td>las_op</td>
<td>sarout</td>
</tr>
<tr>
<td>las2geotiff</td>
<td>SARview</td>
</tr>
<tr>
<td>las2jpeg</td>
<td>SEASAT data</td>
</tr>
<tr>
<td>las2ppm</td>
<td>Seismic events</td>
</tr>
<tr>
<td>Lava flows</td>
<td>Shuttle imaging radar mission</td>
</tr>
<tr>
<td>Level zero data</td>
<td>Signal-to-noise ratio</td>
</tr>
<tr>
<td>line file</td>
<td>Simulated interferogram</td>
</tr>
<tr>
<td>Look angle</td>
<td>SIR-B data</td>
</tr>
<tr>
<td>IZ2RAW_FLYWHEEL</td>
<td>Slant range to ground range</td>
</tr>
<tr>
<td></td>
<td>Spaceborne interferometry</td>
</tr>
<tr>
<td></td>
<td>Spatial resolution</td>
</tr>
<tr>
<td>makeddr</td>
<td>98–99</td>
</tr>
<tr>
<td>Map projection</td>
<td>Speckle</td>
</tr>
<tr>
<td>matchFile</td>
<td>sr2gr</td>
</tr>
<tr>
<td>matrixFile</td>
<td>stats</td>
</tr>
<tr>
<td>meta2ddr</td>
<td>Surface movements</td>
</tr>
<tr>
<td>metadata</td>
<td>System parameters</td>
</tr>
<tr>
<td>Missions</td>
<td>Temporal decorrelation</td>
</tr>
<tr>
<td>Shuttle imaging radar mission</td>
<td>Terrain height</td>
</tr>
<tr>
<td>ml 104–5</td>
<td>Tethered satellite system</td>
</tr>
<tr>
<td>Multibaseline technique</td>
<td>Tools</td>
</tr>
<tr>
<td>4, 8, 11</td>
<td></td>
</tr>
<tr>
<td>Multifrequency approach</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Ocean currents</td>
<td></td>
</tr>
<tr>
<td>Offset</td>
<td></td>
</tr>
<tr>
<td>9, 11</td>
<td></td>
</tr>
<tr>
<td>Orbits</td>
<td></td>
</tr>
<tr>
<td>Ascending orbit</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Crossed orbits</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Descending orbit</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Precise orbits</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>out_dem_seeds</td>
<td></td>
</tr>
<tr>
<td>177</td>
<td></td>
</tr>
<tr>
<td>p2c</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td></td>
</tr>
<tr>
<td>Phase</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Phase filtering</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Phase unwrapping</td>
<td></td>
</tr>
<tr>
<td>8, 10, 21–22</td>
<td></td>
</tr>
<tr>
<td>Baseline refinement</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Phase filtering</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>phase_filter</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td></td>
</tr>
<tr>
<td>Pixel size</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Polar research</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Polar stereographic projection</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Precise orbits</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Processing</td>
<td></td>
</tr>
<tr>
<td>16–23</td>
<td></td>
</tr>
<tr>
<td>projprm</td>
<td></td>
</tr>
<tr>
<td>108–10</td>
<td></td>
</tr>
<tr>
<td>propagate</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td></td>
</tr>
<tr>
<td>Reference DEM</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Reference ellipsoid</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>refine_base</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td></td>
</tr>
<tr>
<td>refine_offset</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td></td>
</tr>
<tr>
<td>remap</td>
<td></td>
</tr>
<tr>
<td>114–15</td>
<td></td>
</tr>
</tbody>
</table>

**ASF software tools – Index**

---

137
## Tools (continued)

<table>
<thead>
<tr>
<th>Tool</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit_quadratic</td>
<td>86</td>
</tr>
<tr>
<td>fit_warp</td>
<td>87</td>
</tr>
<tr>
<td>geocode</td>
<td>88–89</td>
</tr>
<tr>
<td>igram</td>
<td>90</td>
</tr>
<tr>
<td>las_op</td>
<td>91–92</td>
</tr>
<tr>
<td>las2geotiff</td>
<td>93</td>
</tr>
<tr>
<td>las2jpeg</td>
<td>94</td>
</tr>
<tr>
<td>las2ppm</td>
<td>95</td>
</tr>
<tr>
<td>lz2raw_flywheel</td>
<td>96–97</td>
</tr>
<tr>
<td>makeddr</td>
<td>98–99</td>
</tr>
<tr>
<td>metadata</td>
<td>101–3</td>
</tr>
<tr>
<td>ml</td>
<td>104–5</td>
</tr>
<tr>
<td>p2c</td>
<td>106</td>
</tr>
<tr>
<td>phase_filter</td>
<td>107</td>
</tr>
<tr>
<td>proiprm</td>
<td>108–10</td>
</tr>
<tr>
<td>propagate</td>
<td>111</td>
</tr>
<tr>
<td>refine_base</td>
<td>112</td>
</tr>
<tr>
<td>refine_offset</td>
<td>113</td>
</tr>
<tr>
<td>remap</td>
<td>114–15</td>
</tr>
<tr>
<td>resample</td>
<td>116</td>
</tr>
<tr>
<td>reskew_dem</td>
<td>117</td>
</tr>
<tr>
<td>resolve</td>
<td>118</td>
</tr>
<tr>
<td>sarin</td>
<td>119</td>
</tr>
<tr>
<td>sarout</td>
<td>120–21</td>
</tr>
<tr>
<td>sr2gr</td>
<td>122</td>
</tr>
<tr>
<td>stats</td>
<td>123–24</td>
</tr>
<tr>
<td>trim</td>
<td>125</td>
</tr>
<tr>
<td>trim_slc</td>
<td>126</td>
</tr>
<tr>
<td>zeroify</td>
<td>127</td>
</tr>
<tr>
<td>Topographic mapping</td>
<td>3, 4, 10</td>
</tr>
<tr>
<td>TOPSAR data</td>
<td>12</td>
</tr>
<tr>
<td>trim_slc</td>
<td>126</td>
</tr>
<tr>
<td>Universal Transverse Mercator (UTM) projection</td>
<td>22</td>
</tr>
<tr>
<td>SARview</td>
<td>139–41</td>
</tr>
<tr>
<td>XID</td>
<td>143–48</td>
</tr>
<tr>
<td>Volcanic hazards</td>
<td>12</td>
</tr>
<tr>
<td>Critical baseline</td>
<td>8</td>
</tr>
<tr>
<td>Data quality</td>
<td>5</td>
</tr>
<tr>
<td>Differential interferometry</td>
<td>9</td>
</tr>
<tr>
<td>Multifrequency approach</td>
<td>11</td>
</tr>
<tr>
<td>Phase difference</td>
<td>6</td>
</tr>
<tr>
<td>XID</td>
<td>143–48</td>
</tr>
<tr>
<td>zeroify</td>
<td>127</td>
</tr>
</tbody>
</table>
APPENDIX A: SARVIEW

For the visualization of SAR images a tool named SARview has been developed. SARview is a Tcl/Tk and C program that can view LAS images (the internal ASF image format) as well as CEOS SAR images.

The main window of SARview (Figure 17) consists of a menu bar with the three main menu items File, View and Tools. A vertical tool bar is on the left with the tools for zooming, point information and area selection in the shape of a rectangle or a polygon.

The File menu item allows the user to open LAS images as well as CEOS images. It looks for standard LAS extensions such as .img, .amp, .phase, .dem and .ht. For CEOS images generally the extension .D or .dat are used. Any image can be exported to JPEG format.

Figure 17: Main window of SARview
Figure 18: Menu structure of SARview

The View menu item provides more specific information about the particular image. The user can zoom in and out in the main window.

The Selection Info only contains information once an area, either a rectangle or a polygon, is selected. For a selected area the following parameters are given:
- the number of pixels contained in the selected region,
- the length of the area’s edge, and
- the coordinates (in x and y) for the polygon points.

The Image Info provides the name and location of the image file, its data type, the zoom factor used for the display in the main window and the range of the pixels values. The values that can be selected for the Blow-Up Size and the Blow-Up Zoom are summarized in Table 2.

<table>
<thead>
<tr>
<th>Size</th>
<th>Zoom factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 x 200</td>
<td>0.5 x</td>
</tr>
<tr>
<td>300 x 300</td>
<td>1 x (default)</td>
</tr>
<tr>
<td>500 x 500</td>
<td>4 x</td>
</tr>
<tr>
<td>800 x 800</td>
<td>10 x</td>
</tr>
</tbody>
</table>

Table 2: Selection values for the size and zoom factor of the blow-up window

The Tools menu item provides the means to analyze the displayed image in more detail. SARview allows the manipulation of brightness and contrast, and calculates the histogram of the image.

The other four tools can be accessed either via the menu bar or via the tool bar on the left of the main window.
The *Blow Up* option opens a blow-up window with the settings specified in *Blow-Up Size* and *Blow-Up Zoom* in the *View* menu item.

The *Point Info* option opens a point information window. It extracts the information for a particular point selected in the image. The following information is provided:

- the line and sample of the selected point,
- the pixel value,
- the coordinates (in x and y) in a particular map projection if given, and
- the coordinates in geographical coordinates (latitude and longitude in decimal form as well as in degrees, minutes and seconds).

The *Select Rectangle* option takes the first point selected with the left mouse button as the first corner point (upper left or lower right) of the rectangle. The second corner point can be adjusted iteratively.

The *Select Polygon* option takes each left mouse button click as a new point in the polygon.
This page is intentionally blank
APPENDIX B: XID

XID is an X window system application which allows the user to display, interactively manipulate and store image and image related information. XID handles LAS images (the internal ASF image format) as well as generic binary image files.

Figure 19: XID menu window

XID consists of two separate windows, viz. a main display window and a menu bar (Figure 19) with seven menu items File, Tools, View, Display, Graphics, Setup and Info. The entire menu structure is summarized in Table 3.

<table>
<thead>
<tr>
<th>File</th>
<th>Tools</th>
<th>View</th>
<th>Display</th>
<th>Graphics</th>
<th>Setup</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load LAS</td>
<td>Cursor</td>
<td>Full Res</td>
<td>Flicker</td>
<td>GOFs</td>
<td>Layout</td>
<td>Configuration</td>
</tr>
<tr>
<td>Non-LAS Exit</td>
<td>Histo</td>
<td>Save</td>
<td>Initialize</td>
<td>Fill</td>
<td>Preferences</td>
<td>Help</td>
</tr>
<tr>
<td></td>
<td>Power spectrum</td>
<td>Zoom</td>
<td>Map</td>
<td>Mask</td>
<td>Mouse</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Profile</td>
<td></td>
<td>Adjust</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Load Mapping</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Piece-Wise Linear</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Pseudocolor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Save Mapping</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Show</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Menu structure of XID

The File menu allows the user to load images and exit the application.

The Load image option provides a regular pop-up window for the selection of files. Furthermore, the size of image can be specified as well as a certain band combination using standard LAS syntax: <image name>(start line, start sample, number of lines, number of samples: band 1, band 2, band 3).

Example: Load /3dsar/akdem/desc/100/e2_2960_amp.img(1,1,512,512:1). Note that for the selection of subsets the coordinates for the first line and sample are defined as (1,1).
The *Load LAS* image option reads the required image information from the data descriptor record, whereas the *Load Non-LAS* image option requires further input. Apart from the header bytes (number of bytes to skip before the image) the number of bands as well as the number of lines and samples for each band need to be provided. As data types byte, integer, long integer and float values are supported. Bands can be read in the band sequential or band interleaved format.

The **Tools** menu provides the means to analyze the displayed image in detail.

The **Cursor** option allows the interactive analysis of coordinate information (Figure 20).

![Figure 20: Cursor tool of XID](image)

Image and file coordinates are displayed as 3x3 matrices of gray level values. They can be displayed as projection and geographical coordinates as well as unscaled gray level values. The cursor can be moved using the mouse or the arrow keys. The arrow keys move the cursor one pixel in the respective direction. Using the Shift key in combination with the arrow keys moves the cursor ten pixels in the corresponding direction. The gray values can be displayed as original values or as gray values with the current mapping applied. They correspond to either the X/Y dimension or the Z dimension (single band mode) and are updated in automatic fashion or by user interaction clicking the right mouse button. The slider indicates from which band the information is displayed.

The **Histo** option generates a distributional histogram for the currently displayed image (Figure 21).

For each band the mean value, the standard deviation as well as the minimum and maximum pixel value is calculated. The percentage of pixels with a certain gray value can be determined by moving the cursor over the histogram graph.
The histogram can be calculated
- for the currently displayed image, along a line or an area within a defined polygon;
- for all bands of the entire currently displayed image (X/Y) as well as for a line (X/Z) or a column (Y/Z) indicated by the current cursor position;
- as a distributional or a cumulative histogram;
- using the original or the currently mapped image values.
Specific pixel values and/or pixel value ranges can be masked out.

![Figure 21: Histogram tool of XID](image)

The *Power spectrum* option estimates the power spectrum using two different methods, viz. the periodogram method and the maximum entropy method (Figure 22).

The maximum entropy and the periodogram are displayed plotting the power spectral density (PSD) versus the normalized frequency values. For each spectrum the start and end frequency and the number of evaluated points can be chosen. The maximum entropy method also requires the definition of the order used for the calculation. As dimension the power spectrum is calculated in a line, a column or a single point can be defined. The plot can be updated automatically or manually by clicking on a particular point in the image. The points can plotted as a continuous line or as discrete points.
The **Profile** option allows the user to interactively plot profiles of the displayed image (Figure 23).

The profiles can be calculated for a line (X), a column (Y) or a single point (Z). A marker line represents the current sample (X) or the current line (Y) in the profile. The plot can be updated automatically or manually by clicking on a particular point in the image. The points can be plotted as a continuous line or as discrete points.

---

**Figure 22:** *Power spectrum tool of XID*

**Figure 23:** *Profile tool of XID*
The View menu allows the user to display the image in full resolution, to save the current view and change the zoom options.

The Full res option determines the size of the full resolution window (256x256, 512x512, full screen or user defined), whose center point is selected by a left mouse click in the image. It indicates the current full resolution window size and whether its outline is shown in the main image window.

The Save option allows the user to save the main image window, the full resolution window as well as the zoom window either in its entirety or its visible portion.

The Zoom option defines the interpolation method (nearest neighbor, bilinear or cubic convolution) used for resampling of the image data, the use of a point or a continuous zoom and the zoom factor (between 2 and 16) used for the display. The zoom area is selected using the center mouse button.

The Display menu allows the user to define various display options.

The Flicker option allows the user to view a specific list of images in sequence. Image that have been previously saved using the Show option (described below) can be selected and added to the flicker list. The flicker process can be run automatically with the time delay defined using the rate slider. Alternatively, the user can display the flicker list in steps one by one.

The Initialize option provides the user with the capability to clear specified memory and graphic planes and to initialize the display mapping to a linear mapping.

The Map option defines the mapping function used for displaying the image. The user can interactively adjust the brightness and contrast of the displayed image. There are piecewise linear, pseudocolor, interactive, logarithmic and exponential mapping functions available. Any mapping function can be loaded from and saved to disk.

The Show option allows the user to select and save images and image combinations to be displayed in the three color channels. This save list can also be used for the Flicker option described previously. A double click with the left mouse button on an image plane provides further information such as the file name, the window size and the band number.

The Graphics menu allows the user to create and work with graphics overlay files (GOFs).

The GOFs option has all the tools needed to manipulate graphic overlay files. As type of graphics, data points, lines, rectangles, polygons and annotation text are defined.
The *Fill* option defines the area and the way the filling function is applied. As areas, the user selected or all polygons can be selected. The filling function can be applied either inside the defined area or outside. The image can be either updated or a new image can be created. The filling values can be either constant or refer to mapping list that is piece-wise linear or individually renumbered. The filling can be applied to the main window as well as to the full resolution window.

The *Mask* option allows the user to create up to four graphics masks of a specified intensity value and/or a value range. There are eight predefined mask colors.

The *Setup* menu is used to change the memory *Layout* and the user *Preferences* with respect to the cursor color and the number of image colors (200 or 256) used for displaying the images.

The *Info* menu provides general information about the display *Configuration* (the current version of XID, the color depth of display, the use of pseudo color versus direct color and the width and height of the display), links to the online *Help* (requires the NCSA Mosaic browser) and gives information about various *Mouse* functions.
APPENDIX C: FILE FORMATS

.000  –  complex binary level zero swath data

The complex binary level zero swath data is a telemetry data file. It contains frame headers and auxiliary headers. The file is usually accompanied by a chop file (.chop), a parameter file (.par) and a processing log file (.log).

.amp  –  amplitude image

The complex SAR image can be stored either in its real and its imaginary part (.cpx) or as an amplitude image (.amp) as well as a phase image (.phase).

The amplitude image is stored as a flat binary image in floating point format. The accompanying data descriptor file (.ddr) contains further information such as the image dimensions.

.chop  –  chop file

The chop file provides the information on how the level zero CEOS frames can be created from the level zero swath data, so that adjacent frames overlap by an amount of ten percent. This file is not used for any SAR processing by the ASF tools. The format of the chop file is ASCII.

Format:
<number of SAR lines in each scene>
<size of overlap portion between scenes>
scene {
    <first satellite frame number in the scene>
    <last satellite frame number in the scene>
    <number of SAR lines in the frame>
}
[scene …]

Example:
num_scene_lines: 29000
num_overlap_lines: 4096

scene {
    start_index: 24
    end_index: 885074
    number_lines: 29000
}

scene {
    start_index: 760085
    end_index: 1645135
}
ASF software tools – Appendix C: File formats

.number_lines: 29000

scene {
    start_index: 1520146
    end_index: 2405196
    number_lines: 29000
}

...

scene {
    start_index: 12161000
    end_index: 13046050
    number_lines: 29000
}

scene {
    start_index: 12921061
    end_index: 13806111
    number_lines: 29000
}

scene {
    start_index: 13072818
    end_index: 13957896
    number_lines: 29000
}

.cpx – complex binary data in ASF internal format

The complex SAR image can be stored either in its real and its imaginary part (.cpx) or as an amplitude image (.amp) as well as a phase image (.phase). The real and imaginary part are stored as pixel interleaved floating point values in a flat binary image. Further information such as image dimensions is stored in the accompanying data descriptor file (.ddr).

.D – new CCSD data file

The computer compatible signal data (CCSD) represents an ASF internal raw data file format. The header information from the telemetry data is extracted from the original data and stored in the leader file. The extension .D indicates that the CCSD data file has been created after September 1996.

.dat – old CCSD data file

The old CCSD data file format (.dat) differs only slightly from the new CCSD data file format (.D). The header information from the telemetry data is extracted from the original data and stored in the leader file. The extension .dat indicates that the CCSD data file was created before September 1996.
The data descriptor file contains all information about a data set (dimensions, data type, geolocation and map projection) that is required to visualize and combine SAR data processed with the ASF software tools. It is stored in a binary format and follows the LAS data descriptor file structure. This way all images can be visualized using tools that can handle the LAS format.

**Format:**

```
<image name>
<number of lines><number of samples><number of bands><data type>
[...]
<name of map projection>
<zone>
<geodetic datum>
<projection parameters>
[...]
<corner coordinates>
[upper left corner, upper right corner, lower left corner, lower right corner]
<sample spacing><line spacing>
<projection units>
[...]
```

**Example:**

```plaintext
IMAGE NAME:e1
NL:36520   NS:4800   NB:1   DTYPE:REAL*4
LAST MODIFIED: DATE: TIME: SYSTEM:ieee-std
PROJ. CODE:(0)GEOGRAPHIC
ZONE CODE:0   Valid:INVALID
DATUM CODE:0   Valid:INVALID
PROJ. PARM:      Valid:INVALID
  A: 0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00
  B: 0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00
  C: 0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00
  D: 0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00
  E: 0.00000000000000E+00  0.00000000000000E+00  0.00000000000000E+00
CORNER COOR:      Valid:INVALID
ULcorner: 0.00000000000000E+00  0.00000000000000E+00
URcorner: 0.00000000000000E+00  0.00000000000000E+00
LLcorner: 0.00000000000000E+00  0.00000000000000E+00
LRcorner: 0.00000000000000E+00  0.00000000000000E+00
PROJ. DIST: 3.99165415763855E+00  7.90488958358765E+00   Valid:VALID
PROJ. UNITS: meters    Valid:VALID
INCREMENT: 1.00000000000000E+00  1.00000000000000E+00   Valid:VALID
MASTER COOR:1   1

******** image: e1 band: 1 ********

MINIMUM:0.00000000000000E+00   Valid:INVALID
MAXIMUM:0.00000000000000E+00   Valid:INVALID
```

**DATA SOURCE:**
**SENSOR TYPE:**
**CAPT. DIRECTION:**
.dem – digital elevation model file

The extension .dem is generally used for digital elevation models (DEM) in ground range geometry. It does not indicate whether the DEM is geocoded, i.e. whether it has coordinates and a map projection attached to it. This needs to be checked in the accompanying data descriptor file (.ddr). The values in a .dem file are stored in floating point format.

.fmt – data format file

Each sample of SAR data is encoded as two bytes, containing the real and imaginary (I and Q) components of the echo. Window shift and antenna gain control values are provided to correct the antenna gain in order to avoid the saturation of parts of the image. The format of the data format file is ASCII.

Format:
<number of bytes per line of data> <number of header bytes per line>
<mean value of I samples> <mean value of Q samples>
<byte order in file: 'n'=i,q; 'y'=q,i>
[<ignored line>]
[<line1> <window shift (pixels)> <age scale>]
[<line2> <window shift (pixels)> <age scale>]
[...]

Example:
11232 0 ! Bytes per line, bytes per header.
15.500000 15.500000 ! i,q bias
n ! Flip i/q?
! Starting line #, Window Shift (pixels), AGC Scaling
0 4391.938 1

.ht – elevation file

After the unwrapped phase is converted, the elevation is still in the slant range geometry. In order to use the height information for further analysis, it needs to be converted into a ground range DEM (.dem). The elevation file is stored in flat binary format and contains floating point values.
.img – image data file

The extension .img is generally used for any kind of image data. It only implies that the data is stored in a flat binary format. For any further information about the data type or geocoding, the accompanying data descriptor file (.ddr) needs to be consulted.

.in – AISP processing parameter file

The AISP processor parameter file is an ASCII file that contains all relevant parameters used for the processing of a SAR image. It is created while converting level zero STF data or level zero CEOS frame data into ASF internal raw signal data. It is updated as the processing continues. The user can change core parameters such as the range spectrum weight or the bandwidth fractional truncation that cannot be changed by any other input option in the software tools.

Example:
AISP2.5 SAR Processing Parameter File (aisp)
2 ! Debug Flag
0 ! First line (from 0)
1000 ! Number of patches
0 ! i,q byte samples to skip
3320 ! Output lines per patch
0 ! Deskew flag
0 4800 ! 1st range sample, num samples
0.17417745 -0.0000025410 0.0000000001 ! Dopp quad coefs(Hz/prf)
6360563.5 ! Earth Radius
7543.260 ! Body fixed S/C velocity(m/s)
794570.750 ! Spacecraft Height
836610.750 ! Range of first sample
1679.902 ! Pulse Repitition Freq.
8.000000 ! Single look az. res.
5 ! Number of azimuth looks
1.896247E+07 ! Range sampling frequency (Hz)
4.191375E+11 ! Chirp Slope (Hz/s)
3.710000E-05 ! Pulse Length (s)
0.0 ! Chirp extension
0.056565 ! Radar wavelength
0.800000 ! Range spectrum weight
0.000000 0.000000 ! Bandwidth fractional trunc.
0.000000 0.000000 0.000000 0.000000 ! First patch slope,inter range,az
0 0 0 ! Delta per patch slope,inter range,az

.L – new CCSD leader file

The format of the CCSD leader file was changed in September 1996 due to the advanced requirements for the leader file after the Radarsat-1 satellite was launched in November 1995. Changes were made to the structure in order to reflect the enhanced acquisition capabilities with a variety of beam modes (different look angles and resolutions etc.).
.ldr – CEOS leader file

The CEOS leader file contains relevant metadata information to be able to process the SAR data (.raw) it accompanies. It may contain the following records:
- SAR leader file descriptor record
- Data set summary record
- Data quality summary record
- Signal data histogram record
- Processing parameters record
- Platform position data record
- Attitude data record
- Radiometric data record
- Radiometric compensation data record

The contents of the CEOS leader file are stored in binary and ASCII form. The extension .ldr basically indicates a leader file, whether it is leader file for CEOS level zero or CEOS level one or the old CCSD format. The extension of the corresponding binary file answers this question.

.log – level zero STF processing log file

The level zero STF processing log file contains information about the processing of raw data signal from analog into a digital form in the data capture system. It is not relevant for the actual SAR processing as such. However, it provides very useful information for troubleshooting data quality issues. The format of the level zero STF processing log file is ASCII.

Format:
<data capture system (DCS) initialization>
<DCS setup>
<DCS start>
<DCS stop>
  <ss_bit_error_rate>
  <locating time correlation elements>
  <locating state vectors>
  <estimating Doppler centroid parameter>
    <along track Doppler profile at reference range>
    <2D polynomial fitted to observations>
  <Doppler centroid at center_range center_time>

Example:
vxi_setup v2.16
Operating in debug mode
configure_dcs::Constructing Client
configure_dcs::Get intial state of Server
configure_dcs::Built and Sending Command
Command:'Set Parameters' with 8 parameters
  Sent from ID: 0 with response to RespID: 0
Parameter::'/vxdcs/P_AFWS_C001517.down'
Parameter::'/vxdcs/P_AFWS_C001517.par'
Parameter::'256'
Parameter::'2'
Parameter::'0'
Parameter::'0000-00-00:00:00:00'
Parameter::'0'
Parameter::'0000-00-00:00:00:00'
configure_dcs::Command Sent

...

DCS initialized correctly and is ready for capture
Switch -h: OFF
Switch -config: (NULL)
Switch -e: (NULL)
Switch -debug: 0
Switch -id: 268679680
Switch -p: 2051
Switch -i: 192.168.169.131
Switch -d: /vxdcs/P_AFWS_C001517.down
Switch -ip: /vxdcs/P_AFWS_C001517.par
Switch -orb: 0
Switch -sen: (NULL)
Switch -m2s: 264094744
Switch -m2c: 0
Switch -smode: 268483800
Switch -emode: 263620504
Switch -db: ON

20010117030902230: SETUP P_AFWS_C001517 /usr/people/3dsar/L0C/default.data
take
vxi_start v2.16
Operating in debug mode
Creating Client
Sending
Command::'Start Capture' with 0 parameters
Sent from ID: 0 with response to RespID: 0
Received
Command::'Capture Started' with 0 parameters
Sent from ID: 0 with response to RespID: 0
Capture has been started on the DCS
Switch -h: OFF
Switch -config: (NULL)
Switch -e: (NULL)
Switch -db: ON
Switch -id: 0
Switch -p: 2051
Switch -i: 192.168.169.131

20010117030905654: START
vxi_stop v2.16
Operating in debug mode
Creating Client
Got Server State : CapturingSend Command
Command::'End Capture' with 0 parameters
Sent from ID: 0 with response to RespID: 0
Received Command
Command::'Capture Complete' with 0 parameters
Sent from ID: 0 with response to RespID: 0
Capture Complete
Switch -h: OFF
Switch -config: (NULL)
Switch -e: (NULL)
Switch -db: ON
Switch -id: 0
Switch -p: 2051
Switch -i: 192.168.169.131

2001011703133226: STOP
SyncPrep Version 2.8C - Copyright Vexcel Corporation 1999

ss_bit_error_rate:       ERS1 00065537 0000000016777472 0.000000E+00 0.000000E+00
ss_bit_error_rate:       ERS1 00131073 0000000033554688 0.000000E+00 0.000000E+00
ss_bit_error_rate:       ERS1 00196609 0000000050331904 0.000000E+00 0.000000E+00
...

ss_bit_error_rate:       ERS1 13828097 0000003539992832 0.000000E+00 2.882527E-05
ss_bit_error_rate:       ERS1 13893661 0000003556777216 4.885503E-03 6.769676E-05
ss_bit_error_rate:       ERS1 13959176 0000003573549056 8.463383E-03 1.382694E-04

WARNING: Valid sync type ERS located, but actual sensor type could not be identified
INFO: ERS_TCE /usr/people/3dsar/EPHEM/ERS1/E1_22962.tce
19951205201857840::1097938675::0.00390625
INFO: Ephemeris /usr/people/3dsar/EPHEM/ERS1/E1_22962.ORRE
first 1995120520060000, last 1995120521420000
INFO: Block 0, ERS1, O 22962, 19951205203444656 - 19951205203916902

Estimating Doppler Centroid Parameters for prep file /3dsar/TRANS/P_AFWS_C001517.000
ERS_Telemetry::ERS_Telemetry: setting SWST reference to 4568 samples

Data length in seconds to average for each Ambiguity DCEstimate = 2.400000
Number of lines to average for each Ambiguity DCEstimate = 4032
Data length in seconds to average for each Ambiguous DCEstimate = 2.400000
Number of lines to average for each Ambiguous DCEstimate = 4032

Datatake length in seconds determine to be = 272.243198
Minimum ambiguity DCEstimates to perform = 6
Maximum spacing (sec) between Ambiguity DCEstimates = 60.000000
Spacing (sec) between Ambiguous DCEstimate = 15.000000
Number of Ambiguity DCEstimates to be performed = 6
Number of Ambiguous DCEstimates to be performed = 19
SWST change at line 188879 : 4568 <> 4480
shift_pixels = -88
SWST change at line 339982 : 4568 <> 4392
shift_pixels = -176
SWST change at line 415534 : 4568 <> 4304
shift_pixels = -264

Obs Rank Amnr
------------------
0 1 0
1 0 0
2 1 0
3 0 0
4 1 0
5 1 0

2 Observations found with best rank of 0
Choosing observation index 3 as best observation
Obs:0
MLCC(doppler,DAR,ANR,snr):(1927.115477, 1389.105574, 1, 6.568731)
MLBF(doppler,DAR,ANR,snr):(247.213084,458.527198, 0, 1.537383)
Obs:5
MLCC(doppler,DAR,ANR,snr):(2090.114111, 1839.352860, 1, 3.843194)
MLBF(doppler,DAR,ANR,snr):(410.211717,458.527198, 0, 9.019676)

SWST change at line 155079 : 4568 <> 4480 shift_pixels = -88
SWST change at line 274371 : 4568 <> 4392 shift_pixels = -176
SWST change at line 373321 : 4568 <> 4304 shift_pixels = -264

<table>
<thead>
<tr>
<th>Obs</th>
<th>Rank</th>
<th>Amnr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>999</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>999</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>999</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>999</td>
</tr>
<tr>
<td>17</td>
<td>8</td>
<td>999</td>
</tr>
<tr>
<td>18</td>
<td>8</td>
<td>999</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Along Track Doppler Profile at reference range 857646.504527 (m) BEFORE unwrapping
Observation = 0 Doppler = 232.447815
Observation = 1 Doppler = 237.515152
Observation = 2 Doppler = 247.213084
... 
Observation = 21 Doppler = 410.211717
Observation = 22 Doppler = 421.876971
Observation = 23 Doppler = 394.797430
Choosing ambiguity number 0 for Observation Index 14
Obs = 15 df = 6.748153 dt = 5.917606 dfdt = 1.140352 AmnrCorrection = 0
Obs = 16 df = 21.678916 dt = 28.404627 dfdt = 0.763218 AmnrCorrection = 0
Obs = 17 df = 4.908565 dt = 10.651809 dfdt = 0.460820 AmnrCorrection = 0
... 
Obs = 2 df = 2.052382 dt = 13.019209 dfdt = 0.157643 AmnrCorrection = 0
Obs = 1 df = 9.697932 dt = 1.183402 dfdt = 8.194958 AmnrCorrection = 0
Obs = 0 df = 5.067336 dt = 14.202015 dfdt = 0.356804 AmnrCorrection = 0

Along Track Doppler Profile AFTER unwrapping
Observation = 0 Doppler(near) = 265.298138 Doppler(cntr) = 232.447815 Doppler(faar) = 197.584035
Observation = 1 Doppler(near) = 237.061565 Doppler(cntr) = 237.515152 Doppler(faar) = 237.111642
Observation = 2 Doppler(near) = 272.692621 Doppler(cntr) = 247.213084 Doppler(faar) = 219.918475
... 
Observation = 21 Doppler(near) = 430.282566 Doppler(cntr) = 410.211717 Doppler(faar) = 387.819741
Observation = 22 Doppler(near) = 470.224536 Doppler(cntr) = 421.876971 Doppler(faar) = 369.920394
Observation = 23 Doppler(near) = 422.981850 Doppler(cntr) = 394.797430 Doppler(faar) = 363.978098

2DPolynomial fitted to observations
reference_first_dimension : 857646.504527
reference_second_dimension : 1449347820.777995
number_of_coefficients_first_dimension : 4
number_of_coefficients_second_dimension : 3
DAR_doppler = 323.311921 Nominal Doppler = 0.000000 Difference = 323.311921

Max SNR = 9.019676 occurring for observation index 22

Attempting to verify doppler ambiguity with data at reference line 417550 reference pixel 4548

Observation = 0 Asnr = 26.647217 Rsnr = 24.961014 Delta_Ambiguity = 0 doppler = 323.311921

DAR_doppler = 323.311921 Nominal Doppler = 0.000000 Difference = 323.311921

Beam 0::Doppler Centroid at center-range center-time = 323.311921
/3dsar/TRANS/P_AFWS_C001517.000 -status SUCCESS

DATA PROFILE:

Sensor  Rec Status  End Byte  Start Byte  BERR
----------------------------------------------------------------------------
ERS1  REALTIME   003584426367  0000000127  3.424274E-05

SKY_STF_sort Version 2.12 - Copyright Vexcel Corporation 1999

.meta – metadata file

The metadata file is an ASCII file that contains most of the metadata needed for using the ASF software tools. It is created either from the level zero STF parameter file (.par) or from the level zero CEOS leader file (.ldr, .L), while converting the level zero data into ASF internal raw signal data.

Old format:
<parameters used in geolocating the image>
 [...]
<interferometry related parameters>
 [...]
<state vectors>
 [...]
<extra sensor information>
 [...]

New format:
<parameters generally used in remote sensing>
 [...]
<parameters used specifically in SAR imaging>
 [...]
<state vectors>
 [...]
<stats> (optional)
 [...]

158
Old format example:

```
meta_version: 0.9    # ASF-STEP Lab Metadata File.

geo {
    type: S        # Image type: [S=slant range; G=ground range; P=map projected]
    lookDir: R    # SAR Satellite look direction (normally R) [R=right; L=left]
    deskew: 0    # Image moved to zero doppler? [1=yes; 0=no]
    xPix: 7.904889474    # Pixel size in X direction [m]
    yPix: 3.991644988    # Pixel size in Y direction [m]
    rngPixTime: 5.2735745923e-08    # Time/pixel, range (xPix/(c/2.0, or 1/fs) [s]
    azPixTime: 0.0005952782744    # Time/pixel, azimuth (yPix/swathVel, or 1/prf) [s]
    slantShift: 0    # Error correction factor, in slant range [m]
    timeShift: 0.23096585274    # Error correction factor, in time [s]
    slantFirst: 836610.75    # Slant range to first image pixel [m]
    wavelength: 0.056565001607    # SAR Carrier Wavelength [m]
    dopRangeCen: 292.6010437    # Doppler centroid [Hz]
    dopRangeLin: -0.0042686308734    # Doppler per range pixel [Hz/pixel]
    dopRangeQuad: 1.679901942e-07    # Doppler per range pixel sq. [Hz/(pixel^2)]
    dopAzCen: 292.6010437    # Doppler centroid [Hz]
    dopAzLin: 0    # Doppler per azimuth pixel [Hz/pixel]
    dopAzQuad: 0    # Doppler per azimuth pixel sq. [Hz/(pixel^2)]
}

ifm {
    er: 6360563.5    # Local earth radius [m]
    ht: 7155134    # Satellite height, from center of earth [m]
    nLooks: 5    # Number of looks to take from SLC
    orig_lines: 38013    # Number of lines in original image
    orig_samples: 5616    # Number of samples in original image
}

state {
    year: 1996    # Year of image start
    day: 62    # Julian day of the year for image start
    second: 76328.54995    # Second of the day for image start
    number: 3    # Number of state vectors below
    vector {
        time: 0    # Time, relative to image start [s]
        x: -2375522.7151    # X Coordinate, earth-fixed [m]
        y: -1876003.3338    # Y Coordinate, earth-fixed [m]
        z: 6483321.1221    # Z Coordinate, earth-fixed [m]
        vx: -6732.3768539    # X Velocity, earth-fixed [m/s]
        vy: -1693.6516041    # Y Velocity, earth-fixed [m/s]
        vz: -2950.833097    # Z Velocity, earth-fixed [m/s]
    }
    vector {
        time: 30    # Time, relative to image start [s]
        x: -2576417.7159    # X Coordinate, earth-fixed [m]
        y: -1925454.0276    # Y Coordinate, earth-fixed [m]
        z: 6391640.6218    # Z Coordinate, earth-fixed [m]
        vx: -6659.4726129    # X Velocity, earth-fixed [m/s]
        vy: -1602.8469559    # Y Velocity, earth-fixed [m/s]
        vz: -3160.7048761    # Z Velocity, earth-fixed [m/s]
    }
    vector {
        time: 60    # Time, relative to image start [s]
        x: -2775022.3633    # X Coordinate, earth-fixed [m]
        y: -1972161.8845    # Y Coordinate, earth-fixed [m]
        z: 6293709.5734    # Z Coordinate, earth-fixed [m]
        vx: -6579.4726129    # X Velocity, earth-fixed [m/s]
        vy: -1510.8155311    # Y Velocity, earth-fixed [m/s]
        vz: -3367.5023645    # Z Velocity, earth-fixed [m/s]
    }
}
ASF software tools – Appendix C: File formats

} # end vector
} # end of list of state vectors

extra { # begin extra sensor information
  sensor: ERS1 # imaging sensor
  mode: STD # imaging mode
  processor: ASF/AISP/2.90 # Name & Version of SAR Processor
  orbit: 24222 # Orbit Number for this datatake
  bitErrorRate: 0 # Bit Error Rate
  satBinTime: 3044887587 # Satellite Binary Time
  satClkTime: 19960302205327573 # Satellite Clock Time (UTC)
  prin: 1679.9023936 # Pulse Repition Frequency
} # end extra

New format example:
# This file contains the metadata for satellite capture file of the same base name.
# '?' is likely an unknown single character value.
# '???' is likely an unknown string of characters.
# '-999999999' is likely an unknown integer value.
# 'NaN' is likely an unknown Real value.

meta_version: 1.10

general { # Begin parameters generally used in remote sensing
  sensor: RSAT-1 # imaging sensor
  mode: ST5 # imaging mode
  processor: ASF-PGS/PREC/VERS2.16 # Name and Version of Processor
  data_type: BYTE # Type of samples (e.g. REAL64)
  system: big_ieee # System of samples (e.g. big_ieee)
  orbit: 31729 # Orbit Number for this datatake
  orbit_direction: D # Ascending 'A', or descending 'D'
  frame: -999999999 # Frame for this image [-1 if n/a]
  band_number: 0 # Band number; first band is 0
  line_count: 8192 # Number of lines in image
  sample_count: 8192 # Number of samples in image
  start_line: 0 # First line relative to original image
  start_sample: 0 # First sample relative to original image
  x_pixel_size: 12.5 # Range pixel size [m]
  y_pixel_size: 12.5 # Azimuth pixel size [m]
  center_latitude: 63.913975 # Approximate image center latitude
  center_longitude: -145.36534 # Approximate image center longitude
  re_major: 6378144 # Major (equator) Axis of earth [m]
  re_minor: 6356754.9 # Minor (polar) Axis of earth [m]
  bit_error_rate: 8e-07 # Fraction of bits which are in error
  missing_lines: -999999999 # Number of missing lines in data take
} # End general

sar { # Begin parameters used specifically in SAR imaging
  image_type: G # [S=slant range; G=ground range; P=map projected]
  look_direction: R # SAR Satellite look direction [R=right; L=left]
  look_count: 2 # Number of looks to take from SLC
  deskewed: 1 # Image moved to zero doppler? [1=yes; 0=no]
  original_line_count: 8192 # Number of lines in original image
  original_sample_count: 8192 # Number of samples in original image
  line_increment: 1 # Line increment for sampling
  sample_increment: 1 # Sample increment for sampling
  range_time_per_pixel: 7.7358484392e-08 # Time per pixel in range [s]
  azimuth_time_per_pixel: -0.0018946709493 # Time per pixel in azimuth [s]
  slant_range_first_pixel: 966048.4009 # Slant range to first pixel [m]
  slant_shift: 0 # Error correction factor, in slant range [m]
  time_shift: 15.521144417 # Error correction factor, in time [s]
wavelength: 0.0565646
prf: 1270.3251953
satellite_binary_time: ???
satellite_clock_time: ???
dopRangeCen: 4684.90625
dopRangelin: 0.123725
dopRangeQuad: 0
dopAzCen: 4684.90625
dopAzLin: 0
dopAzQuad: 0
}

state {
year: 2001
julDay: 336
second: 57947.433594
vector_count: 3
vector {
time: 0
x: -2283767.3353
y: -2340869.6913
z: 6372816.4062
vx: -6296.9351695
vy: -2622.7211368
vz: -3213.1496582
}
vector {
time: 7.760559082
x: -2332568.1899
y: -2361123.1817
z: 6347672.3633
vx: -6280.5519726
vy: -2595.9219237
vz: -3266.6831055
}
vector {
time: 15.521118164
x: -2381240.1023
y: -2381168.1296
z: 632211.46258
vx: -6263.7309713
vy: -2568.9734392
vz: -3320.0019531
}
}

stats {
min: 0
max: 108
mean: 1.7251530913
rmse: 8.8530471407e-05
std_deviation: 8.8530472067e-05
mask: NaN
}
The null volume directory file is part of the CEOS frame distribution. It is used to terminate a logical data volume. This file has no relevance for the SAR processing. The null volume directory file is 360 bytes long and has binary as well as ASCII parts.

The level zero parameter file contains a large amount of metadata containing information about the sensor (beam, Doppler values, state vectors etc.), the geolocation of the data set as well as the data quality (bit error rate, missing data blocks). It is stored in ASCII format.

Format:

```
[...]
<datatake>
[...]
<ss_block>
[...]
  <local_bit_error_rate>
<prep_block>
[...]
  <sensor>
  [...]
  <beam>
  [...]
  <Doppler centroid parameters>
  <Doppler rate parameters>
<ephemeris data>
[...]
  <state vectors>
  [...]
<location blocks>
<missing blocks>
```

Example:

dcs_version: 2.15f
dcs_id: 1
dcs_file_creation_date: 20010117030859492
dcs_requested_start: 20010117030902588
dcs_satellite: ERS1
dcs_satellite: ERS
dcs_requested_stop:
dcs_start: 20010117030902965
dcs_stop: 20010117031340164
dcs_stop_condition: stop_request
dcs_bit_error_rate: -- No BER Calculated --
dcs_bytes_captured: 3623878656
datatake {
  satellite: UNKNOWN
  instrument: SAR
  tce_UTC: 19500101000000000
  tce_satellite: 0.000000
  tce_corr: 0.000000
  estimated_acq_start: 20200101000000000
  estimated_acq_time: 20.000000
  OrbitNr: 0
  clock_angle: 90.000000
  ellipsoid_name: UNKNOWN

  GHA {
    angle: 0.000000
    date: 19500101000000000
  }
}

js_version: 2.8
js_date: 20010117030857757

ss_block {
  ss_version: 2.8C
  ss_date: 20010117031346651
  block_nr: 0
  sync_type: ERS
  satellite: ERS1
  instrument: SAR
  special_id: None
  transmission_mode: REALTIME
  sync_pattern: faf320
  frame_length: 256
  number_bytes: 3576034816
  number_frames: 13968886
  bit_errors: 7401
  bits_examined: 11761296
  bit_error_rate: 0.000629
  valid_fraction: 0.999831
  invalid_syncs: 3696
  allowed_bit_errors: 1
  flywheel_constant: 29
  IQswap: 0
  invI: 0
  invQ: 0

  local_bit_error_rate {
    ss_bit_error_rate: 00147457 0000000037748992 0.000000E+00
    ss_bit_error_rate: 00278529 0000000071303424 1.828688E-05
    ss_bit_error_rate: 00409601 0000000104857856 2.743033E-05
  ...
  ss_bit_error_rate: 13647873 0000003493855488 9.202002E-06
  ss_bit_error_rate: 13778945 0000003527409920 9.202002E-06
  ss_bit_error_rate: 13910045 0000003560971520 4.489338E-03
}

prep_block {
  prep_version: 2.8C
  prep_date: 20010117032611738
  ss_block: 0
  block_nr: 0

start_byte: 0
number_bytes: 3575966208
number_frames: 13968618
number_lines: 457342
satellite: ERS1
instrument: SAR
beam_sequence: 01
number_of_beams: 1
bit_errors: 7401
bit_error_rate: 0.000629
missing_lines: 0
first_satellite_clock: 1098181060.000000
last_satellite_clock: 1098250755.000000
clock_increment: 0.003906
first_date: 19951205203444656
last_date: 19951205203916902
tce_UTC: 19951205201857840
tce_satellite: 1097938675.000000
tce_corr: 0.000000
estimated_acq_start: 20200101000000000
nominal_look_angle: 21.129204
number_range_samples: 5616
ADC_sampling_frequency: 18962468.000000

state_vector {
x: 5792876.455693
y: 4217456.427326
z: 0.438310
xv: 953.116187
yv: -1323.616953
zv: 7377.132285
Date: 19951205200557121
}
ephemeris_type: RESTITUTED
swath_velocity: 6623.816695

GHA {
  angle: 22.855433
  date: 19951205203444656
}

OrbitNr: 22962
OrbitNr_Date: 19951205203444656
clock_angle: 90.000000

sensor {
sensor_name: ERS1
clock_angle: 90.00000000
nr_temperatures: 0
nr_beams: 1

beam {
  beam_name: ERS1
  nr_of_samples: 5616
  echo_delay: 0.00559208260000
  carrier_freq: 5300000000.00000000
  sampling_freq: 18962468.00000000
  PRF: 1679.90239364
  chirp_rate: 418911637931.03455000
  pulse_length: 0.00003712000000
  polarization: VV
  look_angle: 20.35000000
  incidence_angle: 23.11900000
}
DopplerCentroidParameters {

doppler_centroid_coefficients {
    reference_first_dimension: 857646.504527
    reference_second_dimension: 1449347820.777995
    number_of_coefficients_first_dimension: 4
    number_of_coefficients_second_dimension: 3
    a00: 323.312
    a01: -0.000412394
    a02: 8.12017e-10
    a03: -3.06134e-15
    a10: 0.692892
    a11: -4.44428e-06
    a12: 1.91121e-11
    a13: -1.21493e-16
    a20: -0.000106655
    a21: -3.73235e-08
    a22: 1.8579e-13
    a23: -1.2541e-18
}
reference_range: 857646.504527
reference_date: 19951205203700778
ambiguity_number: 0
MLCC_ambiguity_number_occurence: 2
MLBF_ambiguity_number_occurence: 6
DAR_doppler: 323.311921
Predict_doppler: 0.000000
DAR_confidence: 1.000000
doppler_fit_correlation: 0.823184
doppler_status: SUCCESS
}

DopplerRateParameters {

effective_velocity_coefficients {
    reference_first_dimension: 860429.025906
    reference_second_dimension: 1449347820.777697
    number_of_coefficients_first_dimension: 2
    number_of_coefficients_second_dimension: 2
    a00: 7086.04
    a01: -0.000244044
    a10: 0.0242648
    a11: 8.34305e-08
}
veff: 7086.043134
reference_range: 860429.025906
reference_date: 19951205203700778
autofocus_scale_factor: 1.002319
autofocus_snr: 26.647217
autofocus_suggested_ambiguity_number: 0
autofocus_status: SUCCESS
}
}
ephemeris {
sv_block {
    NrSV: 13

    state_vector {
        x: 5795592.940000
        y: 4213570.200000
        z: 21152.770000
        xv: 934.596270
        yv: -1337.000380
        zv: 7377.145090
        Date: 19951205200600000
    }

    state_vector {
        x: 6666206.850000
        y: 1728084.170000
        z: -1987210.960000
        xv: 2386.716070
        yv: -1091.731450
        zv: 7080.324960
        Date: 19951205214200000
    }

    Attitude {
        yaw: 0.000000
        roll: 0.000000
        pitch: 0.000000
    }

    OrbitNr: 22962
    OrbitNr_Date: 19951205200557133

    GHA {
        angle: 0.000000
        date: 19500101000000000
    }

    Type: RESTITUTED
}

ellipsoid_name: GEM6

location {
    block_nr: 0
    frame_nr: 24
    start_byte: 6144
    satellite_clock: 1098181060.000000
    line_date: 19951205203444656
    first_pixel_ll: 75.543049 -126.082552 0.000000
    last_pixel_ll: 75.974110 -129.765451 0.000000
    SWST_code: 1142
    SWST: 0.00024089348000
    range_gate: 0.00559208260000
    near_range: 838232.09399652
    far_range: 882625.95781551
    platform_altitude: 796130.52320006
    Doppler_centroid: 225.64225719

    DopplerPolynomial {
        reference: 860429.025906
    }
}
number_of_coefficients: 4
a0: 225.642
a1: -0.00049003
a2: 1.57148e-09
a3: -9.76078e-15

location {
  block_nr: 21
  frame_nr: 13957897
  start_byte: 3573221632
  satellite_clock: 1098250755.000000
  line_date: 19951205203916902
  first_pixel_ll: 60.234964 -141.766971 0.000000
  last_pixel_ll: 60.485463 -143.721798 0.000000
  SWST_code: 1076
  SWST: 0.00022697144000
  range_gate: 0.00557816056000
  near_range: 836145.23270053
  far_range: 880539.09651952
  platform_altitude: 793525.92024688
  Doppler_centroid: 414.46960209

  DopplerPolynomial {
    reference: 858342.164610
    number_of_coefficients: 4
    a0: 414.47
    a1: -0.00169949
    a2: 6.76691e-09
    a3: -4.28378e-14
  }
}

missing_data_blocks: 50

missing_data {
  start_frame: 13877248
  start_byte: 3552575488
  missing_bytes: 256
  missing_frames: 1
  missing_lines: 0
}

...
ASF software tools – Appendix C: File formats

.phase – phase image

The complex SAR image can be stored either in its real and its imaginary part (.cpx) or as an amplitude image (.amp) as well as a phase image (.phase).

The phase image is stored as flat binary image in floating point format. Further information can be obtained from the accompanying data descriptor file (.ddr)

.pi – processing information file

The processing information file provides information on the CEOS conversion describing the data set itself as well as its location on the system. The processing information file is stored in ASCII format.

Example:

c eos_block {
  block_nr: 0
  base_name: P_FAAMBD000025
  ss_block: 0
  prep_block: 0
  satellite: RSAT1
  beam_sequence: 18
  scansar_mode:
  number_lines: 30902
  missing_lines: 0
  bit_error_rate: 0.000000
  start_date: 20001001020545399
  stop_date: 20001001020607900
  first_line_first_pixel: 0.000000 0.000000 0.000000
  first_line_last_pixel: 0.000000 0.000000 0.000000
  last_line_first_pixel: 0.000000 0.000000 0.000000
  last_line_last_pixel: 0.000000 0.000000 0.000000
  OrbitNr: 25618
  output_volume: /3dsar/people/mamm/ORBIT/RSAT1/25618/3163.vol
  output_leader: /3dsar/people/mamm/ORBIT/RSAT1/25618/3163.ldr
  output_data: /3dsar/people/mamm/ORBIT/RSAT1/25618/3163.raw
  output_trailer: /3dsar/people/mamm/ORBIT/RSAT1/25618/3163.trl
  output_null: /3dsar/people/mamm/ORBIT/RSAT1/25618/3163.nul
  output_pi: /3dsar/people/mamm/ORBIT/RSAT1/25618/3163.pi
}

.proj – Map projection file

The projection file created by the projprm tool contains a description of the parameters used to define a particular map projection. It defines a projection key that can be used to geocode an image.

Example:

"PROJFILE","Projection parameter file"
"PROJKEY","C","Uniquely identify the defined projection",1,8
"PROJTYPE","I4","Projection type code",1,1
"PROJZONE","I4","Projection zone code",1,1
"PROJUNITS","I4","Units code",1,1
"PROJSPH","I4","Spheroid code",1,1
"PROJPARMS","R8","Projection parameters",1,15;
"UTM6",1,6,2,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0

.raw  – CEOS frame data file / ASF internal raw signal data file

Unfortunately, the extension .raw is used for two different data types, viz. CEOS frame data and ASF internal raw signal data.

CEOS frame data has some header information wrapped around the actual SAR data that has to be stripped off before processing the data. It consists of a data header (one line for ERS data, 16252 bytes for Radarsat data) providing general information such as the image dimensions, and a line header for each line containing information about slant range, geolocation etc. for this particular line. The CEOS frame data is stored in complex format. The real and imaginary parts are each integer (I*2) values.

ASF internal raw signal data is stored in a flat binary format as pixel interleaved I and Q byte values.

.replica – reference function file (Radarsat)

The ASF interferometric SAR processor uses a range pulse replica for compressing the data. For Radarsat data this reference function is externally provided by the .replica file. For ERS data a linear FM chirp is synthesized by aisp. The reference function file is stored in ASCII format.

Format:
<number of samples>
<real part, sample 1> <imaginary part, sample 1>
<real part, sample 2> <imaginary part, sample 2>

Example:
1440
0.0 -1.0
1.2 0.3
...
.stat – statistics file created by stats

The statistics file, calculated by the stats program, provides basic statistical information such as minimum, maximum, mean, rms error and standard deviation for an image file. The statistics can be calculated for the entire image or for a subset. Furthermore, a mask value can be defined that is excluded from the calculation.

Format:
# statistics
<minimum>
<maximum>
<mean>
<rms error>
<standard deviation>
<slope>
<offset>
<upper left line>
<upper left sample>
<lower right line>
<lower right sample>

# histogram
0-7:  <num>  <num>  <num>  <num>  <num>  <num>  <num>  <num>
8-15: <num>  <num>  <num>  <num>  <num>  <num>  <num>  <num>
...

Example:
# Statistics for image file of the same base name
0  # minimum
255  # maximum
27.885725687  # mean
0.002535296453  # rms error
0.0025352964635  # standard deviation
NaN  # masked value (none if NaN)
1  # Slope of line fitting data to [0..255]
0  # Offset of line fitting data to [0..255]
0  # Upper left line of the window statistics were taken in
0  # Upper left sample of the window statistics were taken in
11000  # Lower right line of the window statistics were taken in
11000  # Lower right sample of the window statistics were taken in

# Histogram
0- 7:  35515125  0  0  0  0  16  147  739
8-15:  3070  9308  22164  44686  79071  125531  187433  262746
16-23:  354339  464916  595483  743290  909692  1094517  1287967  1494602
24-31:  2851692  2878119  2878136  2850537  2805043  2743009  2663348  2574927
32-39:  2851692  2878119  2878136  2850537  2805043  2743009  2663348  2574927
40-47:  2479080  2372696  2262862  2153708  2039898  1928401  1817100  1707062
48-55:  1599263  1495404  1393753  1295811  1204431  1116235  1031921  955677
56-63:  878730  810157  742523  679464  624271  570009  520973  477214
64-71:  434435  396350  360129  327341  297662  269558  245953  222841
72-79:  202046  183265  166144  150966  137158  123542  112290  101518
80-87:  92525  83237  75666  68312  61856  56478  50672  46208
.ti – level zero processing information file

The level zero processing information file contains information about the location of level zero STF data. It is stored in ASCII format.

Example:
down_file: /vxdcs/P_FAWS_D013825.down
down_par_file: /vxdcs/P_FAWS_D013825.par

stf_granule {
data_file: /3dsar/ORBIT/RSAT1/25898/P_FAWS_D013825.000
par_file: /3dsar/ORBIT/RSAT1/25898/P_FAWS_D013825.000.par
chop_file: /3dsar/ORBIT/RSAT1/25898/P_FAWS_D013825.000.chop
log_file: /3dsar/ORBIT/RSAT1/25898/P_FAWS_D013825.log
doppler_file:
global_par_file: /3dsar/ORBIT/RSAT1/25898/P_FAWS_D013825.par
ql_gli_image_file:
ql_gli_par_file:
ql_log_file:
ql_image_file:
index_file:
}

.tr1 – CEOS trailer file

The CEOS trailer file is 720 bytes long and contains very limited information about the SAR data such as the processing level. It is not used for the SAR processing of the data.
.vol – CEOS volume descriptor file

The volume descriptor is part of the CEOS frame distribution. It contains a short summary about the processing. It contains very little information except when and where the data has actually been processed. This file has no relevance for the SAR processing. The volume directory file is 360 bytes long and has binary as well as ASCII parts.

avedop – average Doppler file created by estavedop

The average Doppler file created by estavedop contains the constant, linear and quadratic Doppler term that is used by aisp to process the master and slave image to the same Doppler value.

Format:
<Constant Doppler term> <Linear Doppler term> <Quadratic Doppler term>

Example:
0.174177452921867370 -0.000002541001776990 0.000000000133760392

base.00 – baseline file created by resolve

The baseline parameter file created by resolve stores the baseline information as perpendicular and parallel baseline components and their respective rate of change. The original baseline file also contains the temporal baseline in days. This value is not propagated when the baseline is refined.

Format:
<perpendicular baseline> <rate of change in b_{perp}> <parallel baseline> <rate of change in b_{par}> <temporal baseline>

Example:
94.882835  12.870898  -38.806575  -7.358626 1.000001

ctrl – control file created by resolve

The control file created by resolve transfers the offset estimation from fftMatch into a fico compatible format. The original offset of fftMatch is calculated from multilooked amplitude images, whereas fico is applied to single-look complex imagery.

Format:
<offset x (integer)>
<offset y (integer)>
<correlation chip size (32)>
<fraction of a pixel (1)>
Example:
5 195 32 1 4.100000 6.100000

deltas – offset file created by calc_deltas

The offset file created by calc_deltas is used as an input by aisp to perform the resampling of the slave image while carrying out the SAR processing.

Format:
\(<u1> = A*<img1.x> + B\>
\(<v1> = C*<img1.y> + D\>
\(<delu> = E*<img1.x> + F\>
\(<delv> = G*<img1.y> + H\>

Example:
-0.000270 -4.970992 -0.000179 -194.864548
-1.47072e-10 9.45964e-06 -9.05948e-10 9.39941e-07

dem_corr – offset file created by fftMatch

The offset file created by fftMatch contains the offsets in x and y direction of the slave image to the master image and their certainty resulting from an image matching in the Fourier domain. The offset file is used as an initial estimate for a fine co-registration by fico.

Format:
\(<\text{offset x}> <\text{offset y}> <\text{certainty}>\>

Example:
-0.008240 0.810730 74.870575

dem_grid – grid point file created by create_dem_grid

The grid point file created by create_dem_grid serves as an input to fit_plane that transforms the grid points into a mapping plane.

Format:
\(<\text{sar image x}> <\text{sar image y}> <\text{dem x}> <\text{dem y}> <\text{weight}>\)
### Example:

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>-100</td>
<td>1</td>
<td>21496.86031</td>
<td>16710.83630</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>1</td>
<td>19733.09236</td>
<td>16418.61497</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>383</td>
<td>21475.85555</td>
<td>16835.88100</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>383</td>
<td>19712.40709</td>
<td>16543.34100</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>765</td>
<td>21454.82960</td>
<td>16960.92761</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>765</td>
<td>19691.69837</td>
<td>16668.06962</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>1147</td>
<td>21433.78234</td>
<td>17085.97591</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>1147</td>
<td>19670.96608</td>
<td>16792.80058</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>1529</td>
<td>21412.71366</td>
<td>17211.02591</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>1529</td>
<td>19650.21014</td>
<td>16917.53363</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>1911</td>
<td>21391.62344</td>
<td>17336.07667</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>1911</td>
<td>19629.43044</td>
<td>17042.26854</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>2293</td>
<td>21370.51157</td>
<td>17461.12867</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>2293</td>
<td>19608.62687</td>
<td>17167.00507</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>2675</td>
<td>21349.37795</td>
<td>17586.18145</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>2675</td>
<td>19587.79935</td>
<td>17291.74298</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>3057</td>
<td>21328.22245</td>
<td>17711.23478</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>3057</td>
<td>19566.94777</td>
<td>17416.48203</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>3439</td>
<td>21307.04497</td>
<td>17836.28843</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>3439</td>
<td>19546.07204</td>
<td>17541.22199</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>3821</td>
<td>21285.84537</td>
<td>17961.34236</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>3821</td>
<td>19525.17202</td>
<td>17665.96282</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>4203</td>
<td>21264.62351</td>
<td>18086.39660</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>4203</td>
<td>19504.24757</td>
<td>17790.70457</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>4585</td>
<td>21243.37927</td>
<td>18211.45092</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>4585</td>
<td>19483.28663</td>
<td>17915.44695</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>4967</td>
<td>21222.11258</td>
<td>18336.60509</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>4967</td>
<td>19462.32509</td>
<td>18040.18972</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>5349</td>
<td>21200.82332</td>
<td>18461.55889</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>5349</td>
<td>19441.32687</td>
<td>18164.93267</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>5731</td>
<td>21179.51139</td>
<td>18586.61210</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>5731</td>
<td>19420.30388</td>
<td>18289.67554</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>6113</td>
<td>21158.17670</td>
<td>18711.66450</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>6113</td>
<td>19399.25603</td>
<td>18414.41813</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>6495</td>
<td>21136.81914</td>
<td>18836.71588</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>6495</td>
<td>19378.18323</td>
<td>18539.16019</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>6877</td>
<td>21115.43861</td>
<td>18961.76001</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>6877</td>
<td>19357.08540</td>
<td>18663.90150</td>
<td>1.00</td>
</tr>
<tr>
<td>-100</td>
<td>7260</td>
<td>21093.97897</td>
<td>19087.14202</td>
<td>1.00</td>
</tr>
<tr>
<td>5200</td>
<td>7260</td>
<td>19335.90712</td>
<td>18788.96838</td>
<td>1.00</td>
</tr>
</tbody>
</table>

### fico – offset file created by fico

The offset file created by fico contains line and sample locations for offset grid points in the two input images and a signal-to-noise ratio that can be considered a quality value for the estimate. The offset file is used by all fit functions (fit_line, fit_plane, fit_quadratic and fit_warp) that determine transformation parameters for resampling the slave image.

**Format:**

```
<image1 x> <image1 y> <image2 x> <image2 y> <signal-to-noise ratio>
```

**Example:**

```
80    1993 75.00241 1798.04492 3.15
324   1993 319.12146 1798.48804 0.13
```
inVectfile – input vector file used by propagate

The input file for propagating state vectors contains information about the position and velocity of the state vector, the reference and destination point in time as well as the number of steps used for the propagation.

Format:
state vector position in x
state vector position in y
state vector position in z
state vector velocity in x
state vector velocity in y
state vector velocity in z
reference year
reference month
reference day
reference time [s]
destination year
destination month
destination day
destination time [s]
number of steps

Example:
-2464
-2049
6400
-6575
-1939
-1348
1995
12
15
75668
1995
12
15
81668
400
line — regression line file

The regression line file created by fit_line is calculated using a weighted linear first-order least-squares fit and is used by calc_deltas to estimate the offset parameters later used by aisp to resample the slave image. Apart from the actual transformation parameters, it contains the number of points on which the regression is based.

Format:
<img out>.x = <img in>.x + A*<img in>.y + B
<img out>.y = <img in>.x + C*<img in>.y + D
<number of points>

Example:
-0.0002704079592603  -4.9709919028812797  -0.0001789984633230 -194.8645552761054500 301

matchFile — offset file created by fftMatch

The offset file created by fftMatch defines the initial offset between two multilooked amplitude images. It is used by resolve to create an fico compatible input file in order to perform a fine coregistration on the complex imagery.

Format:
<offset x> <offset y> <certainty>

Example:
5.317078 39.004639 88.602478

matrixFile — transformation parameter file created by fit_plane

The transformation parameter file created by fit_plane is calculated using a linear least-squares fit and is used by remap to resample the slave image. Apart from the actual transformation parameters it contains the number of points on which the regression is based.

Format:
<img out>.x = A*<img in>.x + B*<img in>.y + C
<img out>.y = D*<img in>.x + E*<img in>.y + F
<number of points>

Example:
1.0002804168876520  -0.0000059811913303  4.9305921799340053 0.0001795805335320 1.0000025887509492 194.8273171176350047 251
out_dem_seeds – tie point file created by dem2seeds

The tie point file created by dem2seeds (also called by demIFM) from a slant range
digital elevation model contains the location and the height of tie points that are used to
refine the baseline. These tie points are selected as those having minimal local slope.

Format:
<seed x> <seed y> <seed height in meters>

Example:
4795 36495 90.000000
3633 32440 96.000000
3488 32440 96.000000
...
1794 14755 365.000000
3101 8860 245.000000
391 23595 245.000000