Table of Contents

1 Overview

This section is designed to give an overview of how to use the basic Toolbox functions.

What is the ERS SAR Toolbox?

The ERS SAR Toolbox is a collection of software tools, which has been developed to facilitate the use of ERS SAR data. The purpose of the Toolbox is not to duplicate existing commercial packages (e.g. ERDAS or Earthview), but to complement them with functions dedicated to the handling of ESA ERS SAR products.

The Toolbox has a minimum Human Machine Interface and no image display function, but can be integrated within IDL (which is a commercially available image processing package) see Appendix 7.

The Toolbox has been designed to handle the ERS SAR products generated within the ESA ERS ground segment, i.e. by the following processing facilities: ESA Processing and Archiving Facilities (D-PAF, I-PAF, UK-PAF), ESRIN, ESA acquisition stations.

The basic idea

In order to run the Toolbox you type the command:

```
stbx -if file_name.ini
```

where file_name.ini is an ASCII file containing a series of instructions explaining which data to operate on, which operations to perform, and what output to produce. You will need to produce your own .ini files to met your own needs. In practice you will probably do this by editing the .ini files that you already have or by using the HMI system (see below).

Toolbox formats and file extensions

Another key point is that the majority of Toolbox functions operate on data which is in the Toolbox internal format. Therefore it is always necessary to read the data into the Toolbox format using the extraction tools. Any Toolbox operation will produce output data and/or files with one of the extensions listed in section 2.

The HMI

The only function the HMI system has is to help you to create and edit the .ini files. It is perfectly possible to run the Toolbox without the HMI. See Appendix 8 for a description of the HMI system.

The Toolbox contains algorithms to perform the following functions:

Extraction: For generating quicklook and full resolution images from RAW, SLC, SLCI, PRI, GEC, GTC products obtained from the ERS-1 and ERS-2 SAR instruments from various sources and formats; the media shall be EXABYTE or CDROM. Also for decoding the relevant ancillary data and annotating these images.

Data Conversion: For converting between different image formats. For example, between byte and float, power and amplitude and from the internal (Toolbox) image representation into a set of external formats, e.g. TIFF, BIL.

Statistical tool: For calculating global or local statistical parameters from real image data. Also for the computation of the principal components between images.

Resampling tool: For resampling a working image (real or complex) by means of spatial and spectral methods.

Coregistration tool: For automatic co-registration of both real and complex images.

Speckle Filter tool: A Gamma MAP speckle filtering tool for increasing the radiometric resolution of the backscatter image. The filter can be used with real or complex images and has a range of parameters configurable by the user.

Calibration tool: Algorithms for performing various levels of radiometric correction to an input image. Including the generation of an image in which the pixel values represent the backscattering coefficient.

Three Simple Examples: 1 Header Analysis

The purpose of this section is to provide three simple examples of the most basic SAR Toolbox functions. Hopefully this will help to demonstrate the way in which the Toolbox works, so that you can use the Toolbox according to your own needs. In these examples, header information is read from data, a quick look image is generated and a portion of the data is read onto disk.

In order to make a “Quick Look” image or read data onto disc from a CD or tape, it is first necessary to extract information from the header files which accompany the image data. This is done using the Header Analysis function.

To use the Header Analysis function it is necessary to have an ASCII “.ini’, which may look something like this:

```
[HEADER ANALYSIS]
Output Dir = "./"
Input Media Path = "/dev/nrst4"
Input Media Type = "tape"
Product Type = "pri"
Sensor Id = "ers2"
Data Format = "ceos"
Source Id = "itp"
Number Of Volumes = 1
Annotation File = "example"
Header Analysis File = "example"
Dismount Volume = ’N’
```

Let us suppose that this “.ini” file is called header_analysis.ini, then to run this program it is necesary to use the command:

```
stbx -if header_analysis.ini
```

It is useful to examine the contents of the header_analysis.ini file to see the meaning of the differnt instructions. Many further details about the different options available for the Header Analysis .ini files can be found in the main section for this function.

[HEADER ANALYSIS]	This is the name of the function.
Output Dir = "./"	This indicates that the output files will be placed in the directory in which the stbx program is run.
Input Media Path = "/dev/rst1"	This is the path name for the device from which the data will be read. In this case a exabyte tape drive.
Input Media Type = "tape"	The medium on which the data is held. In this case“tape”.
Product Type = "pri"	The type of ERS SARproduct.
Sensor Id = "ers2"	The satellite.
Data Format = "ceos"	The data format.
Source Id = "itp"	The ‘PAF’ at which the data was processed. In this case ‘itp’ indicates the Italian PAF.
Number Of Volumes = 1	The number of tapes. This will usually be 1 unless the data is contained on more than 1 tape.
Annotation File = "example"	The name of the output text file. This will automatically be given the extension .txt.
Header Analysis File = "example"	The name of the output Toolbox format file (input for many other function). This will be given the extension .HAN.
Dismount Volume = ’N’	This indicates that the tape (volume) will not be dismounted after the operation has finished.

Three Simple Examples (continued): 2 the Quick Look function

In order to run the Quick Look function it is necessary to create an ASCII text file which looks something like the following:

```
[QUICK LOOK]
Input Dir = "./"
Output Dir = "./"
Input Media Path = "/dev/rst1"
Input Media Type = "tape"
Header Analysis File = "example.HAN"
Output Quick Look Image = "ql_example"
Output Grid Image = "ql_example_grd"
Number Of Grid Lines = 10,10
Output Image Size = 820, 800
Window Sizes = 3,3
Grid Type = "LATLON"
Grid Drawing Mode = "OVERWRITE"
```

Let us suppose that this “.ini” file is called quick_look.ini, then to run this program it is necesary to use the command:

```
stbx -if quick_look.ini
```

It is useful to examine the contents of the quick_look.ini file to see the meaning of the differnt instructions. Many further details about the different options available for the Quick Look .ini files can be found in the main section for this function.

[QUICK LOOK]	The name of the function.
Input Dir = "./"	The path showing the directory where the required input file will be found. The required input file in this case is “example.HAN” and the path shown here indicates that the file will be found in the directory in which the stbx program is run.
Output Dir = "./"	This indicates that the output files will be placed in the directory in which the stbx program is run.
Input Media Path = "/dev/rst1"	This is the path name for the device from which the data will be read. In this case a exabyte tape drive.
Input Media Type = "tape"	The medium on which the data is held.
Header Analysis File = "example.HAN"	The required input file for this function, which contains information about the data and was created from the Header Analysis function.
Output Quick Look Image = "ql_example"	The output tiff file, with no ‘grid’ super-imposed on it. This file will have the extension .tif.
Output Grid Image = "ql_example_grd"	The output tiff file, which will have a ‘grid’ super-imposed on it. This file will have the extension .tif.
Number Of Grid Lines = 10,10	The number of grid lines (in this case in lat and lon)
Output Image Size = 820, 800	The size of the output image will be 820 rows by 800 columns.
Window Sizes = 11,11	The size of the window used to average the full resolution image to obtain the Quick Look image.
Grid Type = "LATLON"	The grid is drawn along lines of equal lat and lon.
Grid Drawing Mode = "OVERWRITE"	The grid labels overwrite the underlying image.

Three Simple Examples (continued): 3 Full Resolution extraction

In order to run the Full Resolution extraction function it is necessary to create an ASCII text file which looks something like the following:

```
[FULL RESOLUTION]
Input Dir = "./"
Output Dir = "./"Input Media Path = "/dev/rst1"
Input Media Type = "tape"
Header Analysis File = "example.HAN"
Output Image = "full_res_image"
Coordinate System = "LATLON"
Centre = 52.406, 4.470
Size Unit = "KM"
Size = 3.1, 6.3
```

If the ".ini" file is called full_res.ini, then to run this program it is necesary to use the command:

```
stbx -if full_res.ini
```

In order to view the output from the Full Resolution extraction function (i.e. full_res_image.XF?) it will usually be necessary to apply the Gain Conversion function to adjust the dynamic range of the pixel values and convert to an 8 bit image, and then to use the TIFF Conversion tool.

It is useful to examine the contents of the full_res.ini file to see the meaning of the differnt instructions. Many further details about the different options available for the Full Resolution .ini files can be found in the main section for this function.

[FULL RESOLUTION]	The name of the function.
Input Dir = "./"	The path showing the directory where the required input file will be found. The required input file in this case is "example.HAN" and the path shown here indicates that the file will be found in the directory in which the stbx program is run.
Output Dir = "./"	This indicates that the output files will be placed in the directory in which the stbx program is run.
Input Media Path = "/dev/rst1"	This is the path name for the device from which the data will be read. In this case a exabyte tape drive.
Input Media Type = "tape"	The medium on which the data is held.
Header Analysis File = "example.HAN"	The required input file for this function, which contains information about the data and was created from the Header Analysis function.
Output Image = "full_res_image"	The name of the output file, which will be in the Toolbox internal format and which will be given the extension .XTs if the input image was PRI data and will be .XTt if the input image was SLC data.
Coordinate System = "LATLON"	The coordinate system that will be used to extract the region of interest, in this case the lat, lon system, i.e. the same system that was used to draw the grid on the Quick Look image generated previously.
Centre = 52.406, 4.470	The centre of the region from which the region of interest is extracted, in lat, lon coordinates.
Size Unit = "KM"	The system of units used to define the size of the region of interest.
Size = 3.1, 6.3	The size of the region of interest.

2 SAR Toolbox Functions Summary

This section contains a brief summary of all the SAR Toolbox functions.

Extraction from media

1. Header Analysis.
To decode the image header parameters and store them in a plain ASCII file and in the internal Toolbox format. The file in the internal format is a necessary input to the Full Resolution and Quick Look extraction functions.

2. Media Analysis.
To determine from a product on Exabyte media the; number of files in each volume, the number of records in each file and the number of bytes in each record.

3. Quick Look Image Generation.
To generate a reduced resolution version of the full resolution image.

4. Full Resolution Extraction.
To extract a full resolution portion of an image from the media.

5. Portion Extraction.
To extract a full resolution portion of an image from an image already in the Toolbox format.

6. Image Preview.
To select a region of interest from a Quick Look image (i.e. generated using 3). This function is useful to verify that a region of interest is correct, before extracting this region from the full resolution image.

7. Coordinates Retrieving by Example Image.
If a region has been cropped from a Quick Look image using a non-Toolbox tiff-image reading tool, this function makes it possible to establish the coordinates that define the cropped region within the original image.

8. Support Data Ingestion.
This function is required to convert support data (e.g. antenna pattern information or lookup tables for calibration) from an ESA ascii format into the Toolbox internal format.

9. New Product Adding.
This is not really a function in the same sense as the other Toolbox modules. However, this section describes how it is possible to recognise and decode SAR products that were not previously recognised as standard products.

SAR Toolbox Functions Summary (continued)

Data Conversion tools

1. Gain Conversion.
The gain conversion function operates on a floating point or 16 bit integer real image, converting it to an 8 bit image. This tool converts the pixels in an image into a range that makes the image suitable for visualisation. The module is often used prior to the Conversion to TIFF module.

2. Power to Amplitude Conversion.
Converts a power image into an amplitude image.

3. Amplitude to Power Conversion.
Converts an amplitude image into a power image.

4. Linear to dB Conversion.
Converts an amplitude or intensity image with a linear scale into an image in decibel (dB) units.

5. Complex to Amplitude Conversion.
Generates the amplitude modulus from a complex image.

6. Integer to Float Conversion.
Converts a real image from the integer format to the floating point format.

7. Standard TIFF Conversion.
Converts an 8 bit image from the SAR Toolbox internal format to standard TIFF format.

8. Toolbox Format to Band Interleaved (BIL) Conversion (e.g. binary to binary)
Converts three 8 bit images, in the internal Toolbox format, to one standard TIFF RGB image.

9. Ancillary Data Dump.
Generates an ASCII listing of the image annotations relating to an image which is in the internal SAR Toolbox format.

10. Raster Image Import.
Converts an image in RASTER format into the SAR Toolbox format allowing the skipping of both a selectable number of file header bytes number and of line header bytes. The function also generats an ASCII file containing the image size information; this file is compatible with the ERMAPPER “.ers” format)

Statistical Tools

1. Global Statistic.
Calculates a range of statistical parameters for an image or from a region of interest within an image. These statistical parameters are the standard deviation, coefficient of variation, mean value and a histogram of the pixel values.

2. Local Statistic.
For a given input image this function creates a further image based upon the local statistics found within a moving window. The Local Statistics module will give either a ‘mean’, ‘standard deviation’ or ‘coefficient of variation’ image.

3. Principal Component Analysis.
Generates the first and second principal components from a pair of input images.

SAR Toolbox Functions Summary (continued)

Resampling tools

1. Oversampling (Up-Sampling).
This function resamples (up-samples) an image in such a way that the output image contains more pixels than the input image.

2. Undersampling (Down-Sampling).
This function resamples (down-samples) an image in such a way that the output image contains less pixels than the input image.

Coregistration tools

1. Coregistration.
To co-register one or more images with respect to another image. Images can be real or complex.

2. Coherence Generation.
To calculate the degree of coherence between two co-registered complex images.

The Speckle filtering tool

1. Speckle Filter.
The speckle filter tool removes speckle noise from intensity images using the 'Gamma MAP' algorithm. The speckle filter tool operates on real intensity images in the internal SAR Toolbox format.

Calibration tools

1. Backscattering Image Generation.
To convert a power image into backscatter image, i.e. an image that is dependent only on the backscattering properties of the Earth's surface and independent from the satellite and radar parameters.

2. ADC Compensation Image Generation.
This function corrects a power image for the ADC saturation phenomenon. The output may then be used as input in the Backscattering Image Generation function.

3. Gamma Image Generation.
This is to convert a backscatter image (i.e. output from Backscattering Image Generation) into a Gamma image. This is achieved by dividing the backscatter image by the cosine of the satellite incidence angle.

3 SAR Toolbox File Extensions and Internal Format

File Extensions

The SAR Toolbox output file extensions are designed to show which Tool has created them and the type of data that they contain. The extension usually includes two capital letters followed by a lower case letter. The capital letters indicate the Toolbox function, e.g. PA = Power to Amplitude, The lower case letter indicates the format of the pixel data, following the convention:
i = integer 8 bits
f = float 32 bits
c = complex float 32 + 32 bits
s = integer 16 bits
t = integer 16 bits + 16 bits
In addition, text files have the extension .txt and tiff files have the extension .tif.
The Header Analysis function does not produce an image as output, so its internal format consists of 3 capital letters (HAN).

Extraction from media Tools	
1. Header Analysis.	.HAN for internal format, .txt for text file
2. Media Analysis.	.txt
3. Quick Look Image Generation.	.tif
4. Full Resolution Extraction.	.XT? (? shows output format = input format)
5. Portion Extraction.	.XT? (? shows output format = input format)
6. Image Preview.	.tif
7. Coordinates Retrieving by Example Image.	.txt
8. Support Data Ingestion.	.SDf
9. New Product Adding.	Not applicable.
Data Conversion tools	
1. Gain Conversion.	.GCi
2. Power to Amplitude Conversion.	.PAf
3. Amplitude to Power Conversion.	.APf
4. Linear to dB Conversion.	.DBf
5. Complex to Amplitude Conversion.	.CAf
6. Integer to Float Conversion.	.IFf
7. Standard TIFF Conversion.	.tif
8. BIL Conversion.	Not applicable.
9. Ancillary Data Dump.	.txt
10. Raster Image Import.	.RIs (for 16 bit data) .RIt (for 16+16 bit data)
Statistical Tools	
1. Global Statistic.	.txt
2. Local Statistic.	.LSf
3. Principal Component Analysis.	.PCf
Resampling tools	
1. Oversampling (Up-Sampling).	.OVf or .OVc
2. Undersampling (Down-Sampling).	.UNf
Coregistration Tools	
1. Coregistration.	.CRf or .CRc
2. Coherence Generation.	.CHf
The speckle filter tool	
1. Speckle Filter	.SFf
Calibration tools	
1. Backscattering Image Generation.	.BSf
2. ADC Compensation Image Generation.	.ADf
3. Gamma Image Generation.	.GAf

SAR Toolbox Internal Format

The internal format adopted in the SAR Toolbox is called TTIFF, which stands for Tiled Tagged Image File Format. The TTIFF format is a particular form of the commonly used TIFF format. The differences are essentially associated with the name of some image parameters (which, in the TIFF terminology, are called "tags") and with some restrictions in the image organization. An extended discussion of this topic is given in Appendix 9.

The TTIFF files used for the SAR Toolbox internal format are capable of being read by standard display software packages (like XV for UNIX or ULEAD for PC), if the viewer supports the data type contained in the file. For example, it is possible to read integer 8 bit SAR Toolbox internal format images using XV. Integer 8 bit images have the Toolbox file extension (.??i). Where the question marks indicate the module that has been used to produce the image.

Of course, the Standard TIFF conversion module allows any 8 bit SAR Toolbox image to be converted to the standard TIFF format. Internal format data that is not 8 bit, can be converted to 8 bit using the Gain Conversion tool.

Important: When viewing a TIFF image generated by the SAR Toolbox (or the internal format .??i files) using "XV", it is necessary to launch the xv system and load the image from the browser, rather than typing the command xv quicklook.tif.

SAR Toolbox data can also be exported using the BIL conversion tool. This converts one or more integer or floating images in the SAR Toolbox internal format into a band interleaved (BIL) image (i.e. an image written sequentially in a file) which can be seen by an image viewer capable to ingest such data (e.g. ERDAS or ERMAPPER systems). Using the BIL format makes it possible to maintain the data in the source floating point representation, thereby retaining the accuracy of the data.

IDL

SAR Toolbox IDL interface supplies the function READTTF to read SAR Toolbox INTERNAL FORMAT images into IDL.

The READTTF function makes it possible to read all of an image without specifying the starting and ending rows and columns (if this can be done withou overcoming the allowed memory)

This can be done from inside IDL, using

```
img=READTTF(image-name,/ALL)
```

If it is required to read a block, this can be done using

```
img=READTTF(image-name, start-row, start-col, end-row, end-column)
```

A bidimensional array is returned in both cases having a suitable data type set according to the type of data contained in the image.

ERMAPPER

ERMAPPER includes an import menu to load a TIFF image and transform it into its internal format. This option can be also activated via the operating system shell with the following command:

importmany *TIFF-Image-File ERMAPPER-Image-File*

The TIFF grey-level image files are transformed into a one band ERMAPPER file, while both RGB true color and Palette color images are always transformed into three band ERMAPPER image files.

4 Extraction from media

The tools included in this section are:

1. Header Analysis.

To decode the image header parameters and store them in a plain ASCII file and in the internal Toolbox format. The file in the internal format is a necessary input to the Full Resolution and Quick Look extraction functions.

2. Media Analysis.

To determine from a product on Exabyte media the; number of files in each volume, the number of records in each file and the number of bytes in each record.

3. Quick Look Image Generation.

To generate a reduced resolution version of the full resolution image.

4. Full Resolution Extraction.

To extract a full resolution portion of an image from the media.

5. Portion Extraction.

To extract a full resolution portion of an image from an image already in the Toolbox format.

6. Image Preview.

To select a region of interest from a Quick Look image (i.e. generated using 3). This function is useful to verify that a region of interest is correct, before extracting this region from the full resolution image.

7. Coordinates Retrieving by Example Image.

If a region has been cropped from a Quick Look image using a non-Toolbox tiff-image reading tool, this function makes it possible to establish the coordinates that define the cropped region within the original image.

8. Support Data Ingestion.

This function is required to convert support data (e.g. antenna pattern information or lookup tables for calibration) from an ESA ascii format into the Toolbox internal format.

9. New Product Adding.

This function makes it possible to recognise and decode SAR products that were not previously recognised as standard products.

Header Analysis

The Header Analysis function decodes all the header parameters from a product on tape or disc. This information is extracted and stored in a plain ASCII file (extension .txt) and in a file in the Toolbox internal format (extension .HAN). The ASCII file can be examined using a standard text editor to provide useful information about the data. An example of one of these ASCII files is provided in Appendix 1.

Important: The output file in the Toolbox internal format, which has the extension .HAN, is a necessary input to the Quicklook generation and Full Resolution functions.

Example .ini file

```
[HEADER ANALYSIS]
Input Media Path = "/dev/rst1"
Input Media Type = "tape"
Product Type = "pri"
Sensor Id = "ers2"
Data Format = "CEOS"
Source Id = "dep"
Number Of Volumes = 1
Output Dir = "./"
Annotation File = "annot_list"
Header Analysis File = "header_data"
Dismount Volume = 'N'
```

Typical Processing Chain

[HEADER ANALYSIS] -> [QUICK LOOK] -> [FULL RESOLUTION]

Header Analysis Summary Table

Parameter	**Description** **Example**	**Comment**

`Input Media Path`	the path of the media unit;
	for a Unix EXABYTE device use:
	`Input Media Path = "/dev/rst1"`
	for a Unix CDROM device use the entire path to the selected scene (the SAR product CDROM can have multiple scene on it):
	`Input Media Path = "/cdcom/SCENE1/"`
	for a Mac CDROM you have also to insert the CD label e.g.:
	`Input Media Path = "343099:scene1:"`
	for a PC CDROM use:
	`Input Media Path = "D:\scene1\"`
		mandatory parameter

`Input Media Type`	the type of the media, chosen between: • `tape` • `cdrom` `Input Media Type = "tape"`	mandatory parameter
`Product Type`	the type of the product chosen between: • `RAW` • `SLC` • `SLCI` • `PRI` • `GEC` • `GTC` `Product Type = "pri"`	mandatory parameter
`Sensor Id`	the identificator of the sensor chosen between: • `ers1` • `ers2` `Sensor Id = "ers2"`	mandatory parameter
`Data Format`	the format of the product chosen between: • `ceos` • `mphsph` `Data Format = "ceos"`	mandatory parameter

`Source Id`	the PAF or station which have generated the product, chosen between • `ESP` (for ESRIN products) • `DEP` (for DPAF products) • `UKP` (for UKPAF products) • `ITP` (for IPAF products) • `SIS` (for SINGAPORE products) `Source Id = "DEP"`	mandatory parameter
`Number Of Volumes`	the number of exabyte cassettes in which the entire product is subdivided (usually 1) `Number Of Volumes = 1`	mandatory parameter
`Header Analysis File`	the internal format file which will contain all the decoded annotations which shall be used in subsequent processing (an extension `HAN` is automatically added by the system) `Header Analysis File = "header_data"`	mandatory parameter
`Annotation File`	the file which will contain the listing of all the header parameters (an extension `txt` is automatically added by the system) `Annotation File = "annot_list"`	mandatory parameter
`Dismount Volume`	a flag indicating if the media shall be dismounted from the unit at the end of the volume processing; shall be set to `N` when a series of repeated extraction operation are planned on the same cassette, avoiding the waste of time for repeated unit mounting; `Dismount Volume = 'N'` this parameter is ignored (i.e. is assumed `'Y'`) for multi volume processing	optional parameter (if absent is assumed `'Y'`)

Media Analysis

To determine from a product on Exabyte the number of files in each volume, the number of records in each file and the number of bytes in each record.

Important: the Media Analysis function is only possible for data on EXABYTE, it will not work for data on CDROM.

This information extracted by the Media Analysis function is stored in a file called the Media Content Report (Output MCR File) and can be used for the following two purposes:

(1) The media content report contains a clear summary of the product physical structure and can therefore be used to quickly check that the data on the tape corresponds to its label.

(2) If you have a SAR product which does not follow the foreseen CEOS structure (as could happen if it comes from an exotic PAF/Station or if it is damaged), the Media Analysis function allows you to reveal this condition and may eventually give you some ideas on how to modify the FDF files to be able to read the data.

To use this function it is necessary to read the output ASCII MCR file and attempt to evaluate whether the product under consideration is damaged to a degree that makes it un-readable, or whether the unexpected format encountered can be incorporated within the Toolbox framework by the creation of a new FDF file.

Note: An example of an output ASCII MCR file is shown in this manual, in Appendix 2.

Example .ini file

```
[MEDIA ANALYSIS]
Input Media Path = "/dev/rst1"
Number Of Volumes = 1
Output Dir = "./"
Output MCR File = "mcr"
Header Analysis File = "header_data"
Dismount Volume = 'N'
```

Typical Processing Chain

[MEDIA ANALYSIS] -> [HEADER ANALYSIS] -> [QUICK LOOK]

Media Analysis Summary Table

Parameter	Description Example	Comment

`Input Media Path`	the path of the media unit;
	for a UNIX EXABYTE device use:
	`Input Media Path = "/dev/rst1"`
		mandatory parameter

`Number Of Volumes`	the number of exabyte cassettes in which the entire product is subdivided (usually 1) `Number Of Volumes = 1`	mandatory parameter
`Output MCR File`	the name of the file which will contain the media content report (an extension `txt` is automatically added by the system) `Output MCR File = "mcr"`	mandatory parameter

Quick Look Generation

The Quicklook Generation function is used to generate a reduced resolution standard TIFF format version of a full resolution product. This is done by using averaging and subsample operations on the full resolution image.

The full resolution data can be accessed directly from the tape or CD (thus avoiding the creation of large temporary files on the local disk) or from any file that has been created using other Toolbox functions (except for the .tif and BIL files created by this Quick Look function or the BIL Generation function).

Important: When starting with a product on tape or CD, the quicklook extraction function requires the Header Analysis File (extension .HAN) previously generated on the same product which will contain product identifier parameters needed to access the data from the media.

One of the quicklook output image options has superimposed a grid, which helps in the retrieval of the coordinates of the image points. The two coordinate systems in which the grid can be generated are: row, column and latitude, longitude.

The Quick Look image can be displayed in a geometric orientation (option GEOMETRIC, i.e. so that north is up, south is down, west is left and east is right) or in an orientation “as viewed” by the satellite (this is option NORMAL).

Important: When starting from data in a file, the SAR Toolbox internal format image may or may not contain the required ancillary parameters. If these parameters are not present (this will be the case if the image is the output from the Import Raster function of the Data Conversion tool), the grid can be drawn only in row, column coordinates.

The output image (having a selectable size) is stored in standard TIFF format so can be read using any TIFF reader (e.g. “xv” on UNIX, “ulead” on PC, “GIFConverter” on MAC).

Important: When viewing a Quick Look output TIFF image using “xv”, it is necessary to launch the xv system and load the image from the browser, rather than typing the command xv quicklook.tif.

Example .ini file

```
[QUICK LOOK]
Input Media Path = "/dev/rst1"
Input Media Type = "tape"
Input Dir = "./"
Output Dir = "./"
Header Analysis File = "PRIheader_data.HAN"
Output Quick Look Image = "quicklook_nogrid"
Quick Look Presentation = "GEOGRAPHIC"
Output Grid Image = "quicklook"
Number Of Grid Lines = 10,10
Output Image Size = 820, 800
Window Sizes = 11, 11
Grid Type = "LATLON"
Grid Drawing Mode = "OVERWRITE"
```

Typical Processing Chain

[HEADER ANALYSIS] -> [QUICK LOOK]

Quick Look Generation Summary Table

Parameter	**Description** **Example**	**Comment**

`Input Media Path`	the path of the media unit or, when Input Media Type is set to file, the file name of the input internal format image;
	
	for a UNIX EXABYTE device use:
	`Input Media Path = "/dev/rst1"`
	for a UNIX CDROM device use the entire path to the selected scene (the SAR product CDROM can have multiple scene on it) e.g.:
	`Input Media Path = "/cdcom/SCENE1/"`
	for a Mac CDROM you have also to insert the CD label e.g.:
	`Input Media Path = "343099:scene1:"`
	for a PC CDROM use:
	`Input Media Path = "D:\scene1\"`
		mandatory parameter

`Input Media Type`	the type of the media, chosen between: • `tape` • `cdrom` • `file` `Input Media Type = "tape"`	mandatory parameter
`Header Analysis File`	the internal format file containing all the decoded annotations, obtained during the header extraction operation on the same product (with the associated extension `HAN`) `Header Analysis File = "header_data.HAN"`	mandatory parameter when `Input Media Type` is set to `tape` or `cdrom`; ignored for `file` (header data comes from the internal format image annotations)
`Output Quick Look Image`	the name of the TIFF standard file containing the quicklook image, stretched to 8 bit but **without any grid** (an extension `tif` is automatically added by the system) `Output Quick Look Image = "qlk_nogrid"`	mandatory parameter
Quick Look Presentation	the orientation of the output image can be GEOMETRIC, i.e. so that north is up, south is down, west is left and east is right) or in an orientation “as viewed” by the satellite (this is option NORMAL). Quick Look Presentation = "GEOGRAPHIC"	The default (i.e. if no presentation setting is specified) is the GEOMETRIC orientation
`Output Grid Image`	the name of the TIFF standard file containing the quicklook image, stretched to 8 bit but and **containing the grid** (an extension `tif` is automatically added by the system) `Output Grid Image = "quicklook"`	mandatory parameter

`Number Of Grid Lines`	the number of iso-row or (iso-longitude) lines and of iso-column (or iso-latitude) lines of in the grid; the first number refers to iso-row or iso-longitude lines; at least one of number shall be greater than zero `Number Of Grid Lines = 10,10`	mandatory parameter

`Output Image Size`	the number of row, columns in the output quicklook image
	`Output Image Width = 820,800`
	if one of the sizes is set to zero, this means that it shall be automatically computed by the system, using the condition of the equal pixel spacing in the quicklook image, this can be used only on SLC data (on the remeining image products, the specing s are equals); to have a quicklook image with a row size of 500 pixels and equal pixel spacing use:
	`Output Image Size = 500,0`
	to have a quicklook image with a column size of 600 pixels and equal pixel spacing use:
	`Output Image Size = 0,600`
		mandatory parameter

`Window Size`	the number of rows, columns in the moving window used to average the full resolution data during the quicklook creation; use 1 for a pure subsampling and a greater number to obtain a more smoothed image `Window Size = 11,11`	mandatory parameter
`Grid Type`	the type of grid lines chosen between • `ROWCOL` • `LATLON` `Grid Type = "LATLON"`	mandatory parameter
`Grid Drawing Mode`	the drawing mode of the labels used to identify the value of the latitude and longitude at the grid lines crossings, chosen from • `overwrite` (overwrite the underlying image in the label area) • `transparent` (the number contained the label is opaque but the remaining label is transparent) • `none` (no label is written in the image) `Grid Drawing Mode = "overwrite"`	mandatory parameter

`Acknowledge Mount`	this parameter is used to avoid the request to acknowledge of the unit mount during the quicklook generation; to execute a header extraction immediately followed by a quick-look generation (using a unique ".INI" file) set in header analysis section the `Dismount Volume` to '`N`' and set in the quick look section the `Acknowledge Mount` to '`N`': `[HEADER ANALYSIS]` `.....` `Dismount Volume = 'N'` `[QUICK LOOK]` `.....` `Acknowledge Mount = 'N'` this parameter is ignored (i.e. is assumed `'Y'`) in the multi volume processing	optional parameter (if absent is assumed `'Y'`)

Full Resolution Extraction

Function

The Full Resolution Extraction function is used to extract a full resolution image portion from a tape or CD.

The resulting image file will be in the SAR Toolbox internal format and will contain the image pixels plus the various header fields (i.e. the image ancillary data) already obtained with the header extraction operation.

The output image from the Full Resolution Extraction tool will be given an extension .XT?, where the question mark will be replaced by either r, i, s, t, f or c, dependng on the data being read:

r -> when the operation take place on RAW data kept in source media

i -> when the operation take place on 8 bit data generated by the gain convesion task

s -> when the operation take place on PRI, GEC, GTC data kept in source media

t -> when the operation take place on SLC data kept in source media

f -> when the operation take place on internal format data (not generated by gain conversion, oversampling on complex data, coregistration on complex data, raster image import tasks)

c -> when the operation take place on internal format data (generated by oversampling on complex data, coregistration on complex data, raster image import on complex data, tasks)

The image portion (also called AOI, area of interest) can be specified in all the ways described in Appendix 4.

The extracted image has the same pixel format as the source data (no conversion is applied on the pixel values).

Example .ini file

```
[FULL RESOLUTION]
Input Media Path = "/dev/rst1"
Input Media Type = "tape"
Top Left Corner = 0, 0
Bottom Right Corner = 511, 511
Input Dir = "./"
Output Dir = "./"
Header Analysis File = "header_data.HAN"
Output Image = "fullres_image"
```

Typical Processing Chain

[HEADER ANALYSIS] -> [FULL RESOLUTION]

Full Resolution Extraction Summary Table

Parameter	**Description** **Example**	**Comment**

`Input Media Path`	the path of the media unit;
	for a UNIX EXABYTE device use:
	`Input Media Path = "/dev/rst1"`
	for a UNIX CDROM device use the entire path to the selected scene (the SAR product CDROM can have multiple scene on it) e.g.:
	`Input Media Path = "/cdcom/SCENE1/"`
	for a Mac CDROM you have also to insert the CD label e.g.:
	`Input Media Path = "343099:scene1:"`
	for a PC CDROM use:
	`Input Media Path = "D:\scene1\"`
		mandatory parameter

`Input Media Type`	the type of the media, chosen between: • `tape` • `cdrom` `Input Media Type = "tape"`	mandatory parameter
AOI specification	see Appendix 4	optional parameter; if not present, the entire input image is assumed
`Header Analysis File`	the internal format file containing all the decoded annotations, obtained during the header extraction operation on the same product (with the associated extension `HAN`) `Header Analysis File = "header_data.HAN"`	mandatory parameter
`Output Image`	the internal format image which will contain the selected area of interest at full resolution (an extension `XT?` is automatically added by the system) Where the “?” indicates that the output image retains the same format as the input image. `Output Image = "fullres_image"`	mandatory parameter

Portion Extraction

To extract a full resolution portion of an image from an image already in the Toolbox format.
It is much faster to use the Portion Extraction tool to extract sub-images from data that is already in the SAR Toolbox internal format, compared to extracting sub-images from a tape or CD using the Full Resolution extraction function. It may therefore often be useful to use the Portion Extraction function to identify and extract a particular region within an image, after first having used the Full Resolution function to extract a region that is larger than is necessary but which is known to include a feature of interest. In this way it will only be necessary to use the relatively slow Full resolution extraction function once.

The file extension system for the Portion Extraction toll is the same aas that described for the Full Resolution Extraction tool.

The input image must be in the SAR Toolbox internal format image and can be any size (is does not need to correspond to an entire full resolution data set).
The area of interest (AOI) to be extracted can be specified in all of the methods described in Appendix 4, excluding the example image mode but including the polygonal AOI. In the latter case, pixel outside the AOI are set to a zero value. When the input image does not contains the orbital and timing annotations (as happens for images obtained with the Import Raster function of the Data conversion tool) the specification of the AOI using latitude and longitude is obviously not permitted.

Example .ini file

```
[PORTION EXTRACTION]
Input Dir = "./"
Output Dir = "./"
Input Image = "fullres_data.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 511, 511
Output Image = "fullres_portion"
```

Typical Processing Chain

[HEADER ANALYSIS] -> [FULL RESOLUTION] -> [PORTION EXTRACTION]

Portion Extraction Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the input image in internal format `Input Image = "fullres_data.XTs"`	mandatory parameter
AOI specification	see Appendix 4. The example image mode is not permitted and the latitude,longitude is permitted only if the orbital and timing information are present	optional parameter; if not present, the entire input image is assumed
`Output Image`	the name of the output image containing the image portion (an extension “XT?” is automatically added by the system). Where the “?” indicates that the output image retains the same format as the input image. Output Image = "fullres_portion.XTs"	mandatory parameter

Image Preview

To select a region of interest from a Quick Look image (i.e. a .tif image generated using the Quick Look function). This function is useful to verify that a region of interest is correct, before extracting this region from the full resolution image.

The output image is kept in the same standard TIFF format used for the quicklook image so can be read using any TIFF reader (e.g. “xv” on UNIX, “ulead” on PC, “GIFConverter” on MAC).

Example .ini file

```
[IMAGE PREVIEW]
Input Dir = "./"
Output Dir = "./"
Input Image = "quicklook.tif"
Coordinate System = "ROWCOL"
Start Column = 100
Start Row = 100
End Column = 600
End Row = 600
Output Image = "preview"
```

Typical Processing Chain

[HEADER ANALYSIS] -> [QUICK LOOK] -> [IMAGE PREVIEW] -> [FULL RESOLUTION]

Image Preview Generation Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the quicklook image; both the grid or non grid versions can be used `Input Image = "quicklook.tif"`	mandatory parameter
AOI specification	see Appendix 4	mandatory parameter
`Output Image`	the name of the standard TIFF preview image which contains the quicklook of the wanted AOI (an extension `tif` is automatically added by the system) `Output Image = "preview"`	mandatory parameter

Coordinates Retrieving by Example Image

If a region has been cropped from a Quick Look image using a non-Toolbox tiff-image reading tool, this function makes it possible to establish the coordinates that define the cropped region within the original image.

The Coordinates Retrieving function compares two images, the quicklook and a small rectangular portion of it (which is the example image), cropped using an external TIFF reader with such capability. The system then finds the coordinates to the two opposite corners of the example image, expressed in the full resolution row, column coordinate system. This function is useful when the user wants to visually select an AOI using the quicklook image and a TIFF reader and does not want to worry about the coordinates of the AOI. The system then retrieves such coordinates, needed to the full resolution extraction operation. Both the grid or non grid versions of the quicklook image can be used but, of course, a grid quicklook cannot be compared with a non grid example image.
Some care must be taken with external TIFF reader freeware systems with cropping capability due to the presence of bugs and malfunctions. For example, one MacIntosh tool (JPEG View) does not effectively store in the cropped output image a portion of the input one (on the cropped image it can make and undo operation, which means that it store in output the entire input image plus the cropping annotations). The UNIX “XV” tool (version 3.1.0) has shown some problems when cropping a very small image: when the number of columns of the cropped image is below 72, a wrong image is written.
When a wrong example image is given to the Coordinate Retrieving function, a warning message is issued explaining that it will not be possible to retrieve the full resolution coordinates. In such cases, try another cropping tool or use a image processing system (like IDL).

Example .ini file

```
[COORDINATES RETRIEVING]
Input Dir = "./"
Output Dir = "./"
Input Image = "quicklook.tif"
Cropped Tiff Image = "example.tif"
Output Coordinates File = "coords"
```

Typical Processing Chain

[HEADER ANALYSIS] -> [QUICK LOOK] -> [NON SAR TOOLBOX CROPPING TOOL] -> [COORDINATES RETRIEVING]

Coordinates Retrieving Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the quicklook image (both the grid or non grid versions can be used) in standard TIFF format `Input Image = "quicklook.tif"`	mandatory parameter
`Cropped Tiff Image`	an example image cropped from the previous quicklook image, in standard TIFF format `Cropped Tiff Image = "example.tif"`	mandatory parameter
`Output Coordinates File`	the name of the output text file which will contain the row, column coordinates of the Top Right and Bottom Left corners of the example image, expressed in the full resolution coordinate system (an extension `txt` is automatically added by the system) `Output Coordinates File = "coords"`	mandatory parameter

Support Data Ingestion

This function is required to convert support data (e.g. antenna pattern information or lookup tables for calibration) from an ESA ascii format into the Toolbox internal format.

The Support Data Ingestion function converts the external files needed by the SAR Toolbox (antenna pattern, lookup tables for the calibration) into the Toolbox internal format representation. This operation is needed only at the first time the Toolbox is used, or when a change in this data occurs and the older files need to be replaced in the Toolbox. Alternatively the user may want to employ their own antenna pattern or ADC lookup tables.

Example .ini files

The following four ".INI" files show how to transform the two antenna patterns and the two ADC lookup tables from the ESA format (an ASCII file with two columns) into the internal representation (note that these files shall be kept in the './cfg/' directory). This operation must be executed just one time, before any ADC compensation image can be generated and any antenna pattern compensation operation can take place (using the calibration tool):

```
[SUPPORT DATA]
Input Dir = "./cfg/"
Output Dir = "./cfg/"
Input Support Data File = "apers1.dat"
Output Image = "apers1"

[SUPPORT DATA]
Input Dir = "./cfg/"
Output Dir = "./cfg/"
Input Support Data File = "apers2.dat"
Output Image = "apers2"

[SUPPORT DATA]
Input Dir = "./cfg/"
Output Dir = "./cfg/"
Input Support Data File = "adcers1.dat"
Output Image = "adcers1"

[SUPPORT DATA]
Input Dir = "./cfg/"
Output Dir = "./cfg/"
Input Support Data File = "adcers2.dat"
Output Image = "adcers2"
```

Typical Processing Chain

```
[SUPPORT DATA]
```

Support Data Ingestion Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Support Data File`	the external file in ASCII format `Input Support Data File = "ers1_antpat.dat"`	mandatory parameter
`Output Image`	the translated file (an extension `e6` is automatically added by the system) `Output Image = "ers1_antpat"`	mandatory parameter

New Product Adding

This is not really a function in the same sense as the other SAR Toolbox algorithms but instead makes it possible to recognise and decode SAR products that were not previously recognised as standard products.

The New Product Adding function permits the insertion of new format in the set of those recognized by STB. The new format descriptor files shall all be kept in the directory CFG.

To make possible the decoding of a new product, the following operations are required:

- generation of a new section in the file in which all the nominal values of the media structure data are kept
- generation of the FDF file which contains the detailed format structure (following the FDF syntax explained in the next section)
- storing of the FDF file in the configuration directory with a proper naming convention

After these operations the system is capable to accept this new product and treat as a standard products. In the next paragraphs these operations are described in detail.

Nominal Media Structure File description

As already described in the previous paragraph related to the Media Analysis, the value of the following parameters:

- the product format
- the product type
- the sensor Id
- the source PAF

is obtained by a comparison of the media structure (obtained scanning the media, CDRM or EXABYTE) with a file containing the nominal values of this structure for the various products. This file has then as many entries as the number of products which are different each other, in their media structure. Obviously, if two product have the same media structure, there is no way to guess the format, type, sensor and PAF. In the next page is showed a listing showing the nominal media structure file.

A typical entry is the Nominal Media Structure File has the following structure:
PRODUCT PARAMETERS SECTION
VOLUME SECTION *1*
VOLUME SECTION *2*
....
VOLUME SECTION *i*
....
VOLUME SECTION *N*

Within the PRODUCT PARAMETERS SECTION
the structure is:
[*product_type*]

SENSOR:	*sensor_id*
FORMAT:	*format_id*
SOURCE:	*source_id*

where:

- *product_type* can be one of the following three character strings (SLI stands for SLCI product):
  - ⇒ RAW
  - ⇒ SLC
  - ⇒ SLI
  - ⇒ PRI
  - ⇒ GEC
  - ⇒ GTC
- sensor_id can be ERS1 or ERS2 or in general a four character string XXXN; the various sensors are distinguished only by the last character N of their conventional name (so avoid using ERS1 and ENV1 as two different sensors)
- format_id can be CEOS or MPHSPH or in general a n-character string FXX....X, the various formats are distinguished only by the first character F (so avoid using CEOS and CGM as two different formats)
- *source_id* can be:
  - ⇒ ESP (for ESRIN products)
  - ⇒ DEP (for DPAF products)
  - ⇒ UKP (for UKPAF products)
  - ⇒ ITP (for IPAF products)
  - ⇒ SIS (for SINGAPORE products)

  or in general a three character string; all the three character are used to identify the source processing facility

Within a VOLUME SECTION *i*
the structure is:
VOLUME: *i*
FILE SECTION *1*
FILE SECTION *2*
....
FILE SECTION *j*
....
FILE SECTION *M*
where:

- *i* is a integer number (which increases with each volume)

Within a FILE SECTION *j*
the structure is:

FILE: *j* *comparation_flag*
RECORD SECTION *1*
RECORD SECTION *2*
....
RECORD SECTION *k*
....
RECORD SECTION *S*
where:

- *j* is a integer number (which increases with each file)
- *comparation_flag* which can be *Y* or *N*, tell to the system to use or not this information during the comparation of the media structure obtained scanning the media with the values in the nominal media structure file

Each RECORD SECTION *k*
has the following aspect:
RECORD: *recs_number* *rec_lencomparation_flag*
where:

- *recs_number* is the number of sequential records of the same size
- *rec_len* is the record length **in bytes,** common to all the *rec_len* records
- *comparation_flag* which can be *Y* or *N*, tell to the system to use or not this information during the comparation of the media structure obtained scanning the media with the values in the nominal media structure file

An example of one entry of the nominal media structure file is:

```
[PRI]
SENSOR: ERS1
FORMAT: CEOS
SOURCE: ITP
VOLUME: 1
FILE: 1 Y
RECORD:     4   360 Y
FILE: 2 Y
RECORD:     1   720 Y
RECORD:     1  1886 Y
RECORD:     1  1620 Y
RECORD:     1  1046 Y
RECORD:     1 12288 Y
FILE: 3 Y
RECORD:  8201 16012 Y
FILE: 4 Y
RECORD:     1   360 Y
```

This entry describes a ERS1 PRI product of the Italian PAF

- kept on 1 Volume
- organized on 4 files
- file 1 has 4 records of 360 bytes each
- file 2 has
  - ⇒ 1 record of 720 bytes
  - ⇒ 1 record of 1886 bytes
  - ⇒ 1 record of 1620 bytes
  - ⇒ 1 record of 1046 bytes
  - ⇒ 1 record of 12288 bytes
- file 3 has 8201 records of 16012 bytes each
- file 4 has 1 records of 360 bytes

All this information is used during the comparison of the media structure (all flags are Y).
Following these rules, it is possible to add new entries in the Nominal Media Structure File.

Format Description File description

In order to have a tool independent from the product type (PRI, SLC, GEC, etc.) and media (EXABYTE or CDROM) to be de-formatted, a simple description language has been defined to describe the format of each product in a "*Format Description File*" (FDF). In this way, a change of product format does not affect the source code of any module: it is only necessary to properly update the FDF of the correspondent product type. Note that this approach allows to handle in the same manner not only the CEOS format, but also all the positional product-formats (for example, the MPH-SPH product format).

Each *Product Annotation* must be given an annotation name: this name is used to find out the tags related to the annotation itself when inserting it into the internal format image. The format-specifications characteristics of annotations are meaningful to properly compute the size of buffers to be used when reading the product format. These information are all included into the FDF file.

It is allowed the use of variables. Two kinds of variables are available:

- internal variables (named *global variables*);
- *external variables*.

The first ones are useful for two main purposes:

- to store some values that could be necessary later (in future steps, that could come when that values are no more available if they are not stored in temporary and globally accessible variables);
- to access values that are automatically maintained and updated by the de-formatter itself (such as the current date, the number of bytes read from the current record, the current number of processed records, and so on).

There exist a set of pre-defined global-variables the programmer can use. In addition, he can define his own set of global variables and can assign values to them. If useful, they can also be initialized; for this purpose, an apposite optional section of the FDF (named *declaration section*) is provided.

The following sections can be found into a description file:

Declaration Section

It contains the declaration and the initialization of the global variables that must have an initial value. This section should at least contain the variables which describes the various files and record sizes of the product, now described using as example a CEOS PRI. Note that the symbol "@" is part of the variable name and should not be omitted.

Variable	Example value for a CEOS PRI	Comment
`@MAX_F_SZ(1)`	`360`	the maximum record size for the file #1; use 0 for variable length records of a number for a specified record size
`@MAX_F_SZ(2)`	`0`	the maximum record size for the file #2; note that the second file corresponds to the Leader file which has variable length records
`@MAX_F_SZ(3)`	`0`	the maximum record size for the file #3; note that the third file corresponds to the Imagery file which could have variable length records
`@MAX_F_SZ(4)`	`360`	the maximum record size for the file #4
`@F_R(1,1)`	`360`	the actual (fixed) record size of the file #1, record #1
`@F_R(1,2)`	`360`	the actual (fixed) record size of the file #1, record #2
`@F_R(1,3)`	`360`	the actual (fixed) record size of the file #1, record #3
`@F_R(1,4)`	`360`	the actual (fixed) record size of the file #1, record #4
`@F_R(2,1)`	`0`	the record size of the file #2, record #1, set to variable length
`@F_R(2,2)`	`0`	the record size of the file #2, record #2, set to variable length
`@F_R(2,3)`	`0`	the record size of the file #2, record #3, set to variable length
`@F_R(2,4)`	`0`	the record size of the file #2, record #4, set to variable length
`@F_R(2,5)`	`0`	the record size of the file #2, record #5, set to variable length
`@F_R(3,1)`	`0`	the record size of the file #3, record #1, set to variable length
`@F_R(3,2)`	`0`	the record size of the file #3, record #2, set to variable length
`@F_R(4,1)`	`360`	the actual (fixed) record size of the file #4, record #1
`@BITS_PER_SAMPLE`	`16`	the size of each image pixels in bits
`@SAMPLE_PER_PIX`	`1`	the number of layers in the image; use 1 for real images and 2 for complex images
`@IMG_REC_NO`	`0`	initialization of the variable which contain the number of records of the image section
`@IMG_REC_SZ`	`0`	initialization of the variable which contain the record size in bytes of the image records
`@START_LINE`	`0`	initialization of the variable which contain the start line of the image contained in the actual volume
`@END_LINE`	`0`	initialization of the variable which contain the end line of the image contained in the actual volume

Structure Section

It describes the main structure of the product to which the description file is referred. Since a product generally consists of more than one file, this section describes:

- *File Number*: indicates the file ordinal number within the product;
- *Product File Name*: a self explanation string about the file;
- *Product File Code*: the extension used for the file onto the disk;
- *Presence-Type Code* and *Can-Be-Interrupt* flag: these two fields encode how products can span over more than one medium (when necessary because of the product dimensions compared with medium capacity);
- *File Format Code*: the physical format (fixed length, variable length, etc.)
- *File Record Length*: (optional) the default record-size for each file.

File Sections

Each file-section (one section for each file of the actual product) describes the structure of the file it refers to. Since a file can be considered as a sequence of records, a file section consists of a sequence of record sections, each of one describes the logical format of the referred record.

Record Sections

Since a record can be considered as a sequence of fields, each *record section* is a sequence of *field sections*, each of one describes the logical and physical format of the field which it refers to. Note that a *file section* can group several *record sections* into blocks (using the '*repeat construct*'), when iterations on that set of records are required.

Field Sections

A field-section describes how the related field has been filled (when de-formatting) or must be filled (when formatting). Each field-section is composed by the following sub-sections:

- *Field number*: indicates the field ordinal number within the record;
- *Field offset*: indicates the starting bytes as an offset from the beginning of the record;
- *Internal format flag*: it is a flag (the value can be **Y** or **N**) that specifies if the field has to be considered a 'product annotation' (that means, characteristic of the product, independently from its format) or a 'format annotation' (that means, characteristic of the product format but not of the product data itself). Note that all the 'product annotations' are also inserted into the product file in internal format, whereas the 'format annotations' are not included into the product, as they can be obtained from the header listing file, if necessary
- *Align type*: can be **L, R** or **C,** indicating if the field-value has to be respectively left-aligned, right-aligned or centered aligned (respect to the field-size). As a default, numeric fields are generally left-aligned whereas text-fields are right-aligned;
- *Field format*: it describes the formatting convention to be used to decode the field. The allowed formats are:

Strings

⇒ A<u>nnn</u> (where 'nnn' is any integer value), that means a string of <u>nnn</u> characters or even a dummy field of 'nnn' bytes

Integer or floating numbers coded as strings

⇒ I<u>n</u>, that means an integer field expressed using its ASCII representation, with a fixed number (<u>n</u>) of digits. For example, if the 128 integer-number has to be used to fill a I4 field, the four characters will fill the field: '1', '2', '8' and ' ';

⇒ F<u>nn</u>.<u>mm</u>, representing a real number expressed using its ASCII representation, where '<u>nn</u>' and '<u>mm</u>' indicate, respectively, the number of digits before and after the decimal point (for example: F16.7 means a real number of 16 digits, 7 of which after the decimal point).

⇒ D<u>nn</u>.<u>mm</u>, representing a real number in double-precision, expressed using its ASCII representation, where '<u>nn</u>' and '<u>mm</u>' indicate, respectively, the number of digits of the entire number and after the decimal point (for example: D22.10 means a double of 22 digits, 10 of which after the decimal point).

⇒ E<u>nn</u>.<u>mm</u>, representing a real number in exponential format, expressed using its ASCII representation, where '<u>nn</u>' and '<u>mm</u>' indicate, respectively, the number of digits (counting also the 'E' and the + and - sign) of the entire number and after the decimal point.

Integer numbers in binary representation

⇒ B<u>n</u>, that means a binary field of length n (i.e.: a sequence of <u>n</u> bytes), with n equal to **1, 2** or **4** in **NODEC byte ordering**.

⇒ K<u>n</u>, that means a binary field of length n (i.e.: a sequence of <u>n</u> bytes), with n equal to **1, 2** or **4** in **DEC byte ordering**.

Image data

The image data records are recognized by the condition of a REPEAT statement followed by a single record definition. The REPEAT shall use a difference between the two reserved global variables START_LINE and END_LINE. These two variables shall appear defined in the FDF and associated with a couple of fields.

⇒ R<u>n</u>, that means image sample with a pixel having 16 bit unsigned integer format, **in NODEC byte ordering.** The number 'n' is not used in the subsequent processing. If this field starts at a byte position different from 1, the previous bytes are considered as line header and not inserted in the image

⇒ V<u>n</u>, that means image sample with a pixel having 16 bit unsigned integer format, **in DEC byte ordering.** The number 'n' is not used in the subsequent processing. If this field starts at a byte position different from 1, the previous bytes are considered as line header and not inserted in the image

⇒ C<u>n</u>, that means image sample with a pixel having 8 bit signed integer format. The number 'n' is not used in the subsequent processing. If this field starts at a byte position different from 1, the previous bytes are considered as line header and not inserted in the image

- *Tag name*: it is the symbolic name of the field; the system uses this name to retrieve the associated tag (a 16 bits number greater equal than 35001 which is used in the internal format image as field identificator); if a new tag has to be defined, it has to be inserted in the include file **TAGS.H** (kept in the configuration directory **CFG**), which contains all the correspondences between symbolic names and tag numbers; some care must be taken in the usage of names and tag numbers not already defined
- *Future Extension #1*: reserved for future extensions
- *Global Variable Assignment*: used to contain a reference to a global-variable, that will be assigned with the result of the evaluation of the field-value section; it is used to store temporary values that must be used during subsequent de-formatting steps (for example, the number of records contained in a subsequent section, such as the number of state-vectors, number of image lines and so on)
- *Future Extension #2*: reserved for future extensions
- *Future Extension #3*: reserved for future extensions
- *Field Name:* contains the detailed name of the field (as appears in the header listing file obtained in the header decoding function)
- *Units:* contains the measure units of the field (as appears in the header listing file obtained in the header decoding function)
- *Comment:* contains a comment about the field (as appears in the header listing file obtained in the header decoding function)

Naming conventions of the FDF files

The name convention used to identify the various FDF files is mainly based on the restriction to 8 character imposed by the need that the system can run on DOS, which cannot handle long names. The information about the product type, the sensor, the format, the source facility and also the single or multi volume are coded in the following way.

The FDF name is PTYSFSRC.SMV where:

PTY is the three character product type (RAW, SLC, SLI, PRI, GEC, GTC)
S is the single digit number representing the sensor id (1,2)
F is the single character format id (C, M)
SRC is the three character source station (ITP, DEP, UKP, DEP, SIS or a new station)
SMV which is the extension, can be

CFS for single volume products
CFF for multi volume products, first volume
CFC for multi volume products, center volume(s)
CFL for multi volume products, last volume

The various sections of the DFD name shall follow the same values used in the Nominal Media Structure File. As an example, if in the Nominal Media Structure File there is an entry describing a PRI product, coming from a certain ACM station, in a format CEOS, acquired by a ERS2 sensor, split on 1 volume, the corresponding FDF file shall be named: PRI2CACM.CFS.

If was split on two volumes the two FDF shall be named PRI2CACM.CFF and PRI2CACM.CFL, while if was split on three volumes the three FDFs should be PRI2CACM.CFF, PRI2CACM.CFC, PRI2CACM.CFL.

5 Data Conversion tools

The tools included in this section are:

1. Gain Conversion.
The gain conversion function operates on a floating point or 16 bit integer real image, converting it to an 8 bit image. This tool converts the pixels in an image into a range that makes the image suitable for visualisation. The module is often used prior to the Conversion to TIFF module.

2. Power to Amplitude Conversion.
Converts a power image into an amplitude image.

3. Amplitude to Power Conversion.
Converts an amplitude image into a power image.

4. Linear to dB Conversion.
Converts an amplitude or intensity image with a linear scale into an image in decibel (dB) units.

5. Complex to Amplitude Conversion.
Generates the amplitude modulus from a complex image.

6. Pixel to Float Conversion.
Converts a real image from the integer format to the floating point format.

7. Standard TIFF Conversion.
Converts an 8 bit image from the SAR Toolbox internal format to standard TIFF format.

8. BIL Conversion.
Converts three 8 bit images, in the internal SAR Toolbox format, to one standard TIFF RGB image.

9. Ancillary Data Dump.
Generates an ASCII listing of the image annotations relating to an image which is in the internal SAR Toolbox format.

10. Raster Image Import.
Converts an image in RASTER format into the S.AR TOOLbox format (also generating an ASCII file containing the image size information; this file is compatible with the ERMAPPER “.ers” format).

Gain conversion

The gain conversion function operates on a floating point or 16 bit integer real image, converting it to an 8 bit image. This tool will often be used to convert the pixel values in an image into a range that makes the image suitable for visualisation. Therefore, this module is often used prior to the Conversion to TIFF module.

In a SAR image 99% of the image data may have pixel values between 0 and 1000. However, the maximum value of the image may be as great as 30,000. If the conversion from 16 bit to 8 bit is done with a simple scaling, the resultant image will appear all black, except for a very few isolated pixels having values near the maximum. The Gain Conversion module has facilities to allow the conversion to be made in such a way that the image can be successfully visualised.

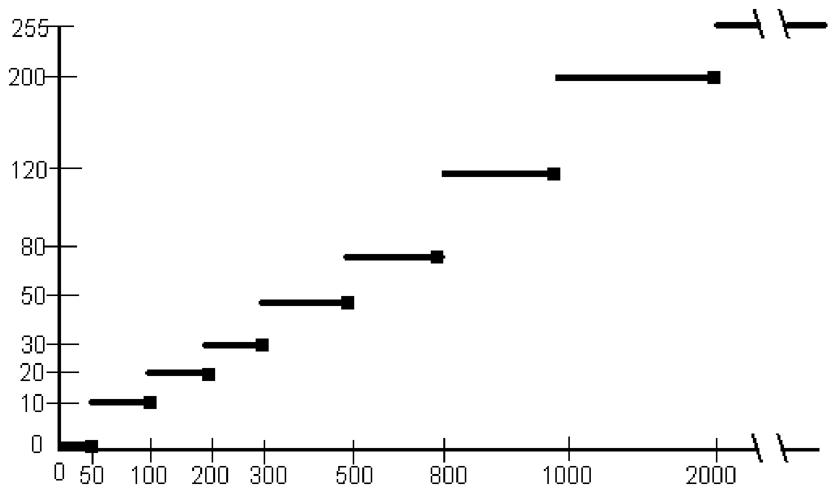
The only Area of Interest which is permitted is the rectangular AOI with corners expressed in the row, col system.
The gain conversion module allows the following three modes of operation. The mode is selected by a parameter specified in the .ini file.

(i) Fixed Gain Conversion. In this mode all of the input image pixel values are divided by a fixed gain value, selected by the user. If the pixel values, after the division, exceed 255 they are saturated at 255. In this way a very simple radiometric stretch scheme can be implemented. The limitation of this method is that the optimum scaling will be dependent on the kind of scene contained in the SAR product. The gain constant will therefore need to be chosen carefully or by using a process of trial and error.

(ii) Variable Gain Conversion. In this mode a linear stretch is performed on those pixels with values that fall between two upper and lower radiometric values, (k_b and k_a respectively). The values k_b and k_a are obtained from the histogram of the image. The user is required to select two percentage values, Min Percentage and Max Percentage (for example, 1% and 99.5%). The two radiometric values k_a and k_b are then calculated, such that Min Percentage (e.g. 1%) of the image pixels have values below k_a and Max Percentage (e.g 99.5%) of the image data have pixel values below k_b. Pixels with values falling between the limits k_a and k_b have a linear stretch applied to them. Pixels with values outside of this range are saturated at 0 and 255.
An optional parameter (Number of Black Levels) can be used to exclude pixels with low values from the calculation, which is used to evaluate the two radiometric values k_b and k_a. If, for example, Number of Black Levels is set to 2.0, then all of those pixels with values lower than 2.0 are not included in the histogram on which the Min Percentage and Max Percentage levels are drawn. The Number of Black Levels parameter allows the computation of a meaningful statistics even for images containing a large portion of pixels near or equal to zero. In particular this is useful for obtaining a good image appearance for GEC or GTC products which contain large regions of black pixels due to the rotation that is applied to these images.

(iii) Look-Up Table Mode. In this mode the user is explicitly required to give the piecewise function that is used to stretch the image. This lookup table should be in the form of an ASCII file contain pairs of numbers which define the way in which the values of the input image are mapped onto the 256 intensity values that are available for the 8 bit output image.
The following examples illustrate the format that is used in the look-up tables.

Look-Up Table Format for Gain Conversion, Example 1



In the figure, above, the x-axis represents the input values and the y-axis represents the output values. The mapping between this input and output is described by the following look-up table:

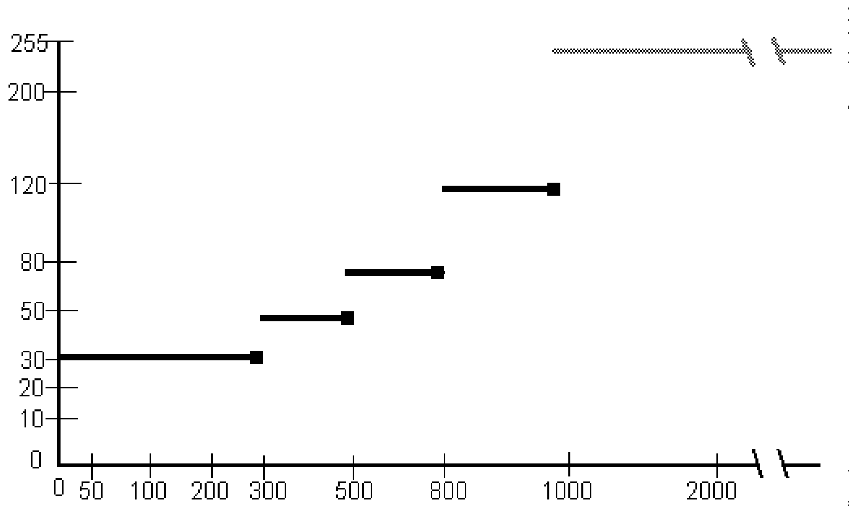
0	50
10	100
20	200
30	300
50	500
80	800
120	1000
200	2000

The first pair of numbers are 0 and 50, this indicates that all pixels in the input image with a value 50 or less are mapped to 0 in the output image. The second pair of numbers are 10 and 100, showing that all pixels in the input image that are between 51 and 100 are mapped to 10 in the output image. The third pair is 20 and 200, indicating that the input pixels with values in the range 101 to 200 are mapped to 20 in the output image.

As can be seen from this example it is not necessary for there to be a separate pair of entries for all of the possible 256 output values.

Also it should be noted that it is not required that the first output value refers to 0 and the last to 255 If the entry corresponding to the output value 255 is missing (as in the example above), the algorithm assumes that all the input values greater than the last present in the lookup shall be set to 255. So, in the example above, all pixels in the input image with values greater than 2000 are set to 200.

Look-Up Table Format for Gain Conversion, Example 2



The lookup corresponding to this figure above is:

30	300
50	500
80	800
120	1000

Notice how the input values greater than 1000 are automatically set to 255.

Example .ini files

The following “.INI” file is an example for the gain conversion with fixed gain.
```
[GAIN CONVERSION]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
Output Image = "fixgain"
```

The following “.INI” file is an example for the gain conversion with variable gain.
```
[GAIN CONVERSION]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
Min Percentage = 0.1
Max Percentage = 99.8
Number of Black Levels = 1.0
Output Image = "vargain"
```

The following “.INI” file is an example for the gain conversion with lookup table.
```
[GAIN CONVERSION]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
User LUT = "lut.dat"
Output Image = "lutgain"
```

Typical Processing Chain

[HEADER ANALYSIS] -> [FULL RESOLUTION]->[GAIN CONVERSION]->[TIFF GENERATION]

Gain Conversion Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the real image in internal format `Input Image = "t1_priimage.XTs"`	mandatory parameter
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
`Scaling Factor`	the constant used to scale the input pixel values `Scaling Factor = 4.0`	mandatory parameter for the fixed gain conversion
`Min Percentage`	the percentage of data at low pixel values which shall be saturated to 0 in the output image, excluding in this count the data having a pixel value less or equal to `Number of Black Levels` `Min Percentage = 0.1`	mandatory parameter for the variable gain stretching mode
`Max Percentage`	the percentage of data which will be scaled linearly, excluding in this count the data having a pixel value less or equal to `Number of Black Levels` and also the data saturated to 0 by the parameter `Min Percentage` `Max Percentage = 99.8`	mandatory parameter for the variable gain stretching mode
`Number of Black Levels`	starting level of the “valid” image pixels; values below this one, are not considered during the histogram evaluation `Number of Black Levels = 1.0`	optional parameter for the variable gain stretching mode (shall be avoided for the remaining conversions)
`User LUT`	the name of the ASCII file containing the lookup table used to stretch the image (in the lookup mode) `User LUT = "lut.dat"`	mandatory parameter for the user lookup table stretching mode
`Output Image`	the name of the output image in internal format containing the image converted in 8 bit pixel modulus data (an extension “`GCi`” is automatically added by the system) `Output Image = "cnvt_image"`	mandatory parameter

Power to amplitude conversion

The power to amplitude function takes the square root of the input image pixel values, thus generating a floating point image containing the amplitude data.

The associated output image annotations which define the pixel type are set to “amplitude”, so that those tools which need amplitude data as input data can first execute a check.

The only AOI which is permitted is the rectangular AOI with corners expressed in row, col system.

Example .ini file

```
[POWER TO AMPLITUDE]
Input Dir = "./"
Output Dir = "./"
Input Image = "power_data.APf"
Output Image = "ampl_data"
```

Power to amplitude Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the input real image in internal format `Input Image = "power_data.APf"`	mandatory parameter
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
`Output Image`	the name of the output image containing amplitude data (an extension "`PAf`" is automatically added by the system) `Output Image = "ampl_data"`	mandatory parameter

Amplitude to Power conversion

The amplitude to power conversion function computes the square of the input image, generating a floating point image containing power data.

The associated output image annotations which define the pixel type are set to “power”, so that those tools which need power data as input data can first execute a check.

The only AOI which is permitted is the rectangular AOI with corners expressed in row, col system.

This function is capable of working with real images but also with complex data, in which case the square modulus is obtained as output. This feature can replace the use of the pipeline between the modulus extraction and the power to amplitude functions, in the case of complex images.

Example .ini file

```
[AMPLITUDE TO POWER]
Input Dir = "./"
Output Dir = "./"
Input Image = "ampl_data.PAf"
Output Image = "power_data"
```

Amplitude to Power Summary Table

Parameter	Description Example	Comment
`Input Image`	the name of the input real image in internal format `Input Image = "ampl_data.PAf"`	mandatory parameter
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
`Output Image`	the name of the output image containing power data (an extension “`APf`” is automatically added by the system) `Output Image = "power_data"`	mandatory parameter

Linear to dB Conversion

The linear to dB conversion function is used to convert an image (an amplitude or an intensity image) from the linear scale to a dB scale.

The AOI which are permitted are the rectangular AOI (with corners expressed in row, col system or in lat, lon) or even the polygonal AOI (in this case the surrounding rectangular AOI is used). No further parameter is needed. Note that to convert a complex image into dB, a modulus extraction shall be executed.

Example .ini file

```
[LINEAR TO DB]
Input Dir = "./"
Output Dir = "./"
Input Image = "pwdata.APf"
Output Image = "data_db"
```

Linear to dB Conversion Summary Table

Linear to dB Conversion Summary Table

Parameter	Description Example	Comment
Input Image	the name of the amplitude or power image in internal format Input Image = "data.PAf" or Input Image = "data.APf"	mandatory parameter
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
Output Image	the name of the output image in dB units (an extension “DBf” is automatically added by the system) Output Image = "data_db"	mandatory parameter

Complex to Amplitude Conversion

The complex to amplitude function works on complex images extracting the modulus, generating a floating point image containing hence amplitude data.

The associated output image annotations which define the pixel type are set to “amplitude”, so that those tools which need amplitude data as input data can first execute a check.

The only AOI which is permitted is the rectangular AOI with corners expressed in row, col system.

Example .ini file

```
[COMPLEX TO AMPLITUDE]
Input Dir = "./"
Output Dir = "./"
Input Image = "slc_data.XTt"
Output Image = "modul_data"
```

Complex to Amplitude Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the complex image in internal format `Input Image = "slc_data.XTt"`	mandatory parameter
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
`Output Image`	the name of the output image containing the modulus data (an extension “`APf`” is automatically added by the system) `Output Image = "modul_data"`	mandatory parameter

Integer to Float Conversion

The integer to float conversion function works on integer images generating the corresponding floating point image. The only AOI which is permitted is the rectangular AOI with corners expressed in row, col system.

Important: Please note that the title of the function in the .ini file is [PIXEL TO FLOAT] rather than [INTEGER TO FLOAT]. This inconsistency will be changed in a later version of the SAR Toolbox.

Example .ini file

```
[PIXEL TO FLOAT]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
Output Image = "float_img"
```

Integer to Float Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the integer image in internal format `Input Image = "t1_priimage.XTs"`	mandatory parameter
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
`Output Image`	the name of the output image containing floating point data (an extension “`IFf`” is automatically added by the system) `Output Image = "float_img"`	mandatory parameter

Standard TIFF Conversion

The standard TIFF conversion function can convert an 8 bit image in the internal SAR Toolbox format into a standard grey-level tiff image which can be read by a general purpose image viewer. Or the standard TIFF conversion function can be used to convert three 8 bit images in the internal SAR Toolbox format into a standard colour tiff image which can be read by a general purpose image viewer.

No AOI is permitted in this conversion. If a single image is specified in input then a grey level TIFF image is generated. If three images are specified in input then a RGB color TIFF image is generated.

Important: If the image viewer "xv" is used to visualise the output from the TIFF conversion module, it may be necessary to first launch the "xv" system and then read-in the image from the "load" menu option, rather than using, for example, the command:

```
xv grey_img.tif
```

This is because of the nature of the TIFF file generated by the Toolbox module.

Example .ini files

The following ".INI" file is an example for single TIFF image conversion:

```
[TIFF GENERATION]
Input Dir = "./"
Output Dir = "./"
Input Images = "vargain.GCi"
Output Image = "grey_img"
```

The following ".INI" file is an example for the RGB TIFF image conversion:

```
[TIFF GENERATION]
Input Dir = "./"
Output Dir = "./"
Input Images = "red.GCi","green.GCi","blue.GCi"
Output Image = "rgb_img"
```

Standard TIFF Conversion Summary Table

Parameter	**Description** **Example**	**Comment**

`Input Images`	the name of a single 8 bit images in internal format
	`Input Images = "vargain.GCi"`
	or of three 8 bit images in internal format which constitute the Red, Green , Blue layers of the output true color image
	`Input Images = "red.GCi","green.GCi", "blue.GCi"`
		mandatory parameter

`Output Image`	the name of the TIFF output image containing a grey scale or a true color RGB image (an extension "`tif`" is automatically added by the system) `Output Image = "tiff_img"`	mandatory parameter

Toolbox Format to Band Interleaved (BIL) Conversion (e.g. binary to binary)

The BIL conversion converts one or more integer or floating images in the SAR Toolbox internal format into a band interleaved (BIL) image (i.e. an image written sequentially in a file) which can be seen by an image viewer capable to ingest such data (e.g. ERDAS or ERMAPPER systems). Using the BIL format makes it possible to maintain the data in the source floating point representation, thereby retaining the accuracy of the data.
No AOI is permitted in this conversion.

A maximum of **10** images can be used as input and it is required that the input images share the same type and sizes. The conversion generates both the output image file and an associated ASCII descriptor containing the main image information. This descriptor can be read by the ERMAPPER system (it has the same structure as the ".ers" files) thus allowing a direct loading of this BIL data. When the data type of the pixel is not allowed by ERMAPPER (e.g. the 8 + 8 bit complex pixels) only the image data is generated (no ".ers" file).

Example .ini files

```
[BIL GENERATION]
Input Dir = ".\
Output Dir = ".\
Delete Input Image = "N"
Input Images = "input.XTs"
Output Image = "output"
```

will convert an integer 16 bit image in the Toolbox internal format into a binary image called output (no extension is added by SAR Toolbox) having the following structure:

```
<line1,pixel1> .... <line1, pixelM>
<line2,pixel1> .... <line2, pixelM>
.....          .....
<lineN,pixel1> .... <lineN, pixelM>
```

where N is the number of rows of input.XTs and M is the number of columns. Each value <lineI,pixelJ> is a signed 16 bits integer. This information can be found in the output.ers file produced for ER Mapper Integration. Note that this (.ers) file (for ER Mapper) is not produced for Complex Unsigned 8 bit Integer and Complex Float image data, while the output is always produced.

The following example makes uses 4 input files:

```
[BIL GENERATION]
Input Dir = "./"
Output Dir = "./"
Input Images = "band1.XTs","band2.XTs","band3.XTs","band4.XTs"
Output Image = "bil_img"
```

BIL Conversion Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Images`	the input image list (can also be) `Input Images = "band1.XTs","band2.XTs","band3.XTs","band4.XTs"`	mandatory parameter
`Output Image`	the name of the BIL output image containing multiband data; (no extension is added by the system but an ASCII descriptor file with the same name and extension "`ers`" is generated) `Output Image = "tiff_img"`	mandatory parameter

Ancillary Data Dump

The ancillary data dump creates an ASCII file containing a listing of the image annotations which are stored in the specified internal format image. The listing includes the following fields:

- a line progressive index
- the annotation name
- the annotation value

and is a fast way to check the value of the image annotations.
The SAR Toolbox system maintains the entire annotation set only for the header file coming from the extraction tools. The remaining tools maintain only the annotations needed (which are considerably less, due to the fact that the most part of the original CEOS or MPHSPH are not used for the SAR Toolbox processing) with the exception of the TIFF and the BIL conversion function which cut all the annotations from the output file.

Example .ini file

```
[ANCILLARY DATA DUMP]
Input Dir = "./"
Output Dir = "./"
Input Image = "cfvr.LSf"
Output File = "dump"
```

Note: The extension .LSf for the Input Image is just one possible example of a file in the internal SAR Toolbox format.

Ancillary Data Dump Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the input image `Input Image = "cfvr.LSf"`	mandatory parameter
`Output File`	the name of the ASCII file containing the annotation listing (an extension “`txt`” is added by the system) `Output File = "dump"`	mandatory parameter

A sample of a file obtained as output from this function is shown in Appendix 3, together with a table to explain the set of the annotations that are maintained by the various STB tools.

Raster Image Import

This function is able to convert from SAR Toolbox internal format into raster format (16 bit for integer or 16+16 bits for complex integer). The data to be converted can be initially present on disk, CD or tape, thus avoiding the need for the user to dump the image using the operating system commands.

Using the Raster Image Import function, it is possible to convert external images in the format required by the SAR Toolbox tools, even for data not belonging to the CEOS or MPHSPH format, but having similar pixel size. Due to the fact that the function operates on pure image data, no annotation is inserted in the output internal format image. Obviously, this limits the number of SAR Toolbox functions which can accept the output of the Raster Image Import function as input. Often the external images will have both a file header section (once for the image) and a line header (for each line), the raster import function is capable to skip both headers, obtaining a output data containing only the image pixels only (instead of a mixture of pixel and header bytes). Even if no direct AOI can be used, using together the following parameters:

- File Header Bytes
- Line Header Bytes
- Number of Rows
- Number of Columns

it is possible to define a rectangular AOI region to be converted.
This function, allowing the direct media access, can be easily used to extract images with corrupted or missing header. Due t

Example .ini file

The following “.INI” file is an example for a real raster image conversion (the parameters are those used to convert a 500 rows by 500 columns portion of a CEOS PRI image file from a EXABYTE tape unit on a Unix machine):

```
[IMPORT RASTER]
Input Dir = "./"
Output Dir = "./"
Input Media Type = "tape"
Input Image = "/dev/rst1"
Media File Skip = 2
Data Type = "2I"
File Header Bytes = 16012
Line Header Bytes = 12
Image Record Length = 16012
Number of Rows = 500
Number of Columns = 500
Swap Bytes = "N"
Output Image = "imported_img"
```

Raster Image Import Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Media Type`	the type of the media, chosen between: • `disk` • `tape` • `cdrom` `Input Media Type = "disk"`	optional parameter (set to `disk` as default)

`Input Image`	when `Input Media Type` is set to `disk` or to `cdrom`, this parameter corrspond to the name of a 2I or complex 2I image in RASTER format; when `Input Media Type` is set to `tape` this parameter shall be set to the device name of the tape unit:
	`Input Image = "external_img.dat"`
	or for a Unix EXABYTE
	`Input Image = "/dev/rst1"`
		mandatory parameter

`Data Type`	the indication that the RASTER data refers to a 16 bit real image
	`Data Type = "2I"`
	or a complex 16+16 bit image
	`Data Type = "Complex 2I"`
		mandatory parameter

`Media File Skip`	the number of files which precede the image file to be converted; these files will be skipped `Media File Skip = 2`	optional parameter (set to 0 by default); this parameter is not used when `Input Media Type` is set to `Disk` or `CDROM`
`File Header Bytes`	the number of byte to skip once at the start of the file; typically these bytes constitutes the file header section before the image data `File Header Bytes = 16012`	optional parameter (if missing 0 is assumed)
`Line Header Bytes`	the number of byte to skip at the start of each image line; typically these bytes constitutes the header section of each line, containing non image data `Line Header Bytes = 12`	optional parameter (if missing 0 is assumed)
`Image Record Length`	the record length of the RASTER image, expressed as number of bytes `Image Record Length = 16012`	mandatory parameter
`Number of Rows`	the number of rows of the input image to be converted `Number of Rows = 500`	mandatory parameter when `Input Media Type` is set to `Tape` optional parameter in the remaining cases (if missing all the rows are read)
`Number of Columns`	the number of columns of the input image to be converted `Number of Columns = 500`	optional parameter (if missing all the columns are read)

`Swap Bytes`	the indication that each byte couple shall be swapped or not before writing in the output file; can be set to `"Y"` to execute the swapping or to `"N"` to left the byte ordering untouched; use `"Y"` when reading a CEOS data (which is stored in a NONDEC format) on a PC machine, but set it to `"N"` if trying to read on the same machine a MPHSPH product (which is stored in DEC format); the PC machine is actually a DEC ordering machine `Swap Bytes = "Y"`	optional parameter, set by default to `"N"`
`Output Image`	the name of the internal format output image (an extension `v8` is automatically added by the system) `Output Image = "imported_img"`	mandatory parameter

6 Statistical Tools

The tools included in this section are:

1. Global Statistic.

Calculates a range of statistical parameters for an image or from a region of interest within an image. These statistical parameters are the standard deviation, coefficient of variation, mean value and a histogram of the pixel values.

2. Local Statistic.

For a given input image this function creates a further image based upon the local statistics found within a moving window. The Local Statistics module will give either a ‘mean’, ‘standard deviation’ or ‘coefficient of variation’ image.

3. Principal Component Analysis.

Generates the first and second principal components from a pair of input images.

The statistical tool operates with images in internal format and is capable of the following functions:

Global Statistic

The global statistic function computes some statistical parameters inside an area of interest (AOI). The statistical parameters are the standard deviation, coefficient of variation, mean value, image max, image min and a histogram of the pixel values. These values are global, i.e. one unique value of a certain statistic parameter is given for the entire AOI.
The AOI can be specified in any way, except the example image mode.

Example .ini file
The following ".INI" file is an example for the global statistics (the input PRI image portion creation is described in section "Extraction from media").

```
[GLOBAL STATISTIC]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 100, 100
Bottom Right Corner = 700, 700
Class Min = 100.0
Class Max = 900.0
Classes Number = 8
Output File = "glostat"Global Statistic Summary Table
```

Global Statistic Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the input real image in internal format `Input Image = "t1_priimage.XTs"`	mandatory parameter
AOI specification	see Appendix 4	optional parameter; if not present, the entire input image is assumed
`Class Min`	a floating number specifying the **second** class of the histogram; the image pixels having a value lower or equal than this, shall all contribute to the first histogram bin `Class Min = 100.0`	mandatory parameter
`Class Max`	a floating number specifying the **next to last** class of the histogram; the image pixels having a value greater or equal than this, shall all contribute to the last histogram bin `Class Max = 900.0`	mandatory parameter
`Classes Number`	an integer number specifying the number of classes of the histogram **minus 2**; the histogram shall contain two class more than this number, the first for all the pixel having a value below `Class Min`, the second for all the pixel having a value greater than `Class Max` `Classes Number = 8`	mandatory parameter
`Output File`	the name of the output ASCII file containing the global statistical data (an extension "`txt`" is automatically added by the system) `Output File = "glostat"`	mandatory parameter

Local Statistic

The local statistic function computes a statistical parameter within a (usually small) window which is allowed to move across an input image. The statistical parameter that is computed for each position of this window is used to build-up an output image, which then contains information about the local statistics of the input image.

The statistical parameters available are the mean, standard deviation or coefficient of variation.

It is possible for the user to specify the size of the moving window in which the statistical parameters are calculated. It is also possible for the user to specify the step sizes that determines the frequency with which the statistical parameter is calculated. These step sizes will also determine the size of the output image.

It is also possible for the user to define the output image size by specifying the Output Image Ratio values along the rows and columns. See the second Example .ini file below.

The area of interest (AOI) for the input image can be specified in any way (except the example image mode), including polygonal AOI. When the moving window is partly or completely outside the AOI no statistics are generated. For more details see the “Statistical tools” chapter of the Algorithmic Specification Document [A3].

Example .ini files

The following “.INI” file is an example for the local statistics (the input PRI image portion creation is described in section “Extraction from media”).

```
[LOCAL STATISTIC]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Filler Value = 0.0
Coordinate System = "ROWCOL"
Top Left Corner = 100, 100
Bottom Right Corner = 500, 500
Output Image Type = "MEAN"
Window Sizes = 5, 5
Window Steps = 2.0, 2.0
Output Image = "t1_locstatmean"
```

In the following example the the output image size is determined by the parameter Output Image Ratio.

```
[LOCAL STATISTIC]
Input Dir = "./"
Output Dir = "./"
Input Image = "pri.XTs"
Output Image = "local"
Output Image Type = "MEAN"
Output Image Ratio = .5, .7
Window Sizes = 3, 5
Top Left Corner = 100, 100
Bottom Right Corner = 299, 399
```

This .ini file will produce an output image “local.LSf” with (299-100+1)*0.5 rows and (399-100+1)*0.7 columns.

Local Statistic Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the input real image in internal format `Input Image = "t1_priimage.XTs"`	mandatory parameter
AOI specification	see Appendix 4	optional parameter; if not present, the entire input image is assumed
`Output Image Type`	the type of local statistical operation: • `MEAN` for evaluating the moving mean • `SDDV` for evaluating the moving standard deviation • `CFVR` for evaluating the moving coefficient of variation `Output Image Type="MEAN"`	mandatory parameter
`Window Sizes`	the size of the moving window used to compute the local statistic; a couple of integer numbers comma separated, the first one referring to the number of rows, the second to the number of columns `Window Sizes = 5, 5`	mandatory parameter
Output Image Ratio	the ratio by which input image is transformed into the output image Output Image Ratio = .5, .7	This parameter is an alternative to the Window Steps parameter
`Window Steps`	the step at which the local statistic window is moved; set to a value different from 1 if you want to sub sample the corresponding statistic image; e.g. if you compute a local statistic on a 100 by 100 image with a 10 by 10 window and a 20 by 20 step, then your output image will be only 5 by 5 pixels wide; if you use a step of 1 by 1 instead you will obtain a full image 91 by 91 pixels (i.e. less than 100 by 100 because of the “window edge” effect) `Window Steps = 2.0, 3.0`	This parameter is an alternative to the Output Image Ratio parameter
`Filler Value`	in case of usage of a polygonal AOI, the output image (always rectangular) could contains zones whose pixels are outside the AOI and shall so be set at a certain value; this parameter permits to choose this value `Filler Value = -1.0`	optional parameter; if not found the filler is set to 0
`Output Image`	the name of the output image in internal format containing the local statistical data (an extension “`s1`” is automatically added by the system) `Output File = "locmean"`	mandatory parameter

Principal Component Analysis

The Principal Component Analysis is used to generate the first and second principal component images from a pair of input images. Only the rectangular AOI specification is possible, with corners expressed in row, col system. The two output images are scaled to avoid negative pixel values.

Example .ini file

```
[PRINCIPAL COMPONENT ANALYSIS]
Input Dir = "./"
Output Dir = "./"
Input Images = "img1.ttf", "img2.ttf"
PCA Output Images = "pc1", "pc2"
```

Principal Component Analysis Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Images`	the name of the input real images in internal format, in a comma separated list `Input Images="img1.ttf", "img2.ttf"`	mandatory parameter
AOI specification	only the Rectangular AOI specifications are permitted (see Appendix 4)	optional parameter; if not present, the entire input images are assumed
`PCA Output Images`	the name of the output images in internal format containing the Principal Component images (an extension "`PCf`" is automatically added by the system) in a comma separated list `PCA Output Images="pc1","pc2"`	mandatory parameter

7 Resampling tools

The tools included in this section are:

1. Oversampling (Up-Sampling).
This function resamples (up-samples) an image in such a way that the output image contains more pixels than the input image.

2. Undersampling (Down-Sampling).
This function resamples (down-samples) an image in such a way that the output image contains less pixels than the input image.

Oversampling (Up-Sampling)

The Oversampling function up-samples a real or complex image (like a SLC product) using the FFT and Zero Pad algorithm. The algorithm takes into account the value of the Doppler Centroid Frequency when padding the azimuth spectrum.

The size of the output image can either be determined by specifying this directly in terms of the pixel dimensions, e.g. Output Image Size. Or, in terms of the row, col ratios by which the input image should be multiplied. These numbers should be greater than 1 (see the second example .ini file below).

Only the rectangular AOI with corners specified in row, col system, are accepted.

Example .ini files
The following “.ini” file is an example for the oversampling for an input SLC image portion.

```
[IMAGE OVERSAMPLING]
Input Dir = "./"
Output Dir = "./"
Input Image = "slcimage.XTt"
Output Image = "oversam"
Top Left Corner = 100,100
Bottom Right Corner = 199,299
Output Image Size = 200,300
```

The output from the following .ini file would have dimensions (299-100+1)*1.5 rows and (399-100+1)*1.1 columns.

```
[IMAGE OVERSAMPLING]
Input Image = "priimage.XTs"
Output Image = "oversam"
Top Left Corner = 100,100
Bottom Right Corner = 299,399
Output Image Ratio = 1.5,1.1
```

Oversampling (Up-Sampling) Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the input real or complex image in internal format `Input Image = "slcimage.XTs"`	mandatory parameter
AOI specification	only the Rectangular AOI specifications are permitted (see Appendix 4)	optional parameter; if not present, the entire input images are assumed
`Output Image Size`	the size of the output image, expressed as number of rows, number of columns; these sizes shall be greater than the input image size; this number permits to control the over-sampling factor `Output Image Size=2000,1500`	This parameter is an alternative to Output Image Ratio
Output Image Ratio	The row, col ratios to determine the size of the output file. These numbers should be greater than 1. Output Image Ratio = 1.5,1.1	This parameter is an alternative to Output Image Size
`Output Image`	the name of the output image in internal format containing the oversampled image (an extension “`OVf or OVc`” is automatically added by the system) `Output Image = "oversam"`	mandatory parameter

Undersampling (Down-Sampling)

The undersampling function down-samples a real image (like a PRI or GEC product) using a kernel, which moves across the input image with a step-size determined by the size of the required output image.

The size of the output image can either be determined by specifying this directly in terms of the pixel dimensions, e.g. Output Image Size. Or, in terms of the row, col ratios by which the input image should be multiplied. These numbers should be less than 1 (see the second example .ini file below).

The kernel can be selected for one of a set of predefined kernels or can be defined by the user. A list of the pre-defined kernels is given here after the ‘example .ini file’.

The kernel file is a simple ASCII file containing rows of coefficients separated by spaces, with each row terminated with a newline (return) character.
There are two modes in which the kernel files can be used:

2D: One method allows 2-dimensional filters to be defined directly. In this case an example of an ASCII file for a 3x3 averaging filter would be:

```
0.1111 0.1111 0.1111
0.1111 0.1111 0.1111
0.1111 0.1111 0.1111
```

This particular example is contained in the pre-defined file usam33.ker.

2 x 1D: The other method makes use of two 1-dimensional filters to effectively synthesise a 2-dimensional filter. This method involving two 1-dimensional filters can be much faster than the direct use of a 2-dimensional filter. For example, for a 11 rows by 11 columns filter, this filtering can be five times faster than a conventional 2-dimensional filtering.

The two 1-dimensional filters that would produce the equivalent filter to the 3x3 averaging filter shown above would be defined by the following ASCII file:

```
0.3333 0.3333 0.3333
0.3333
0.3333
0.3333
```

The SAR Toolbox system automatically discovers the correct way to interpret the kernel depending on the layout of the ASCII file.

As another example, consider the following 4x3 filter:

```
1.0 2.0 3.0 4.0
2.0 4.0 6.0 8.0
3.0 6.0 9.0 12.0
```

This could be synthesised by two 1-dimensional filters described by the following ASCII representation:

```
1.0, 2.0, 3.0, 4.0
1.0
2.0
3.0
```

Example .ini file

The following .ini file shows the function working on a PRI image with an output image size defined in terms of pixels.

```
[IMAGE UNDERSAMPLING]
Input Dir = "./"
Output Dir = "./"
Filter File Name = "usam33.ker"
Input Image = "priimage.XTs"
Top Left Corner = 100, 200
Bottom Right Corner = 399, 599
Output Image = "undersam"
Output Image Size = 200, 200
```

The following .ini file shows the function working on a PRI image with an output image size defined in terms of ratios.

```
[IMAGE UNDERSAMPLING]
Input Dir = "./"
Output Dir = "./"
Filter File Name = "usam33.ker"
Input Image = "pri.XTs"
Top Left Corner = 100, 100
Bottom Right Corner = 299, 399
Output Image = "undersam"
Output Image Ratio = .5, .7
```

This would give an output file of size (299-100+1)*0.5 rows and (399-100+1)*0.7 columns.

The following table describes the library of preset kernels.

Kernel file name	**Comment**
edd_3_3.ker	3x3 Edge Detect
ede_3_3.ker	3x3 Edge Enhance
lop_3_3.ker	3x3 Low Pass
hip_3_3.ker	3x3 High Pass
hor_3_3.ker	3x3 Horizontal
ver_3_3.ker	3x3 Vertical
sum_3_3.ker	3x3 Summary
edd_5_5.ker	5x5 Edge Detect
ede_5_5.ker	5x5 Edge Enhance
lop_5_5.ker	5x5 Low Pass
hip_5_5.ker	5x5 High Pass
hor_5_5.ker	5x5 Horizontal
ver_5_5.ker	5x5 Vertical
sum_5_5.ker	5x5 Summary
edd_7_7.ker	7x7 Edge Detect
ede_7_7.ker	7x7 Edge Enhance
lop_7_7.ker	7x7 Low Pass
hip_7_7.ker	7x7 High Pass
hor_7_7.ker	7x7 Horizontal
ver_7_7.ker	7x7 Vertical
sum_7_7.ker	7x7 Summary

Undersampling (Down-Sampling) Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the real input image in internal format `Input Image = "t1_priimage.XTs"`	mandatory parameter
AOI specification	only the Rectangular AOI specifications are permitted (see Appendix 4)	optional parameter; if not present, the entire input image is assumed
`Filter File Name`	the name of the ASCII file which contains the coefficients of the filter used to subsample the input image; the file names of the preset filters are shown in “” `Filter File Name="usam55.ker"`	mandatory parameter
`Output Image Size`	the size of the output image, shall be lower than the input image size; this number permits to control the undersampling factor `Output Image Size=20,15`	This parameter is an alternative to Output Image Ratio
Output Image Ratio	The row, col ratios to determine the size of the output file. These numbers should be between 0.0 and 1.0. Output Image Ratio = 0.7,0.5	This parameter is an alternative to Output Image Size
`Output Image`	the name of the output image in internal format containing the undersampled image (an extension “`UNf`” is automatically added by the system) `Output Image = "undersam"`	mandatory parameter

8 Coregistration tools

The tools included in this section are:

1. Coregistration.
To co-register one or more images with respect to another image. Images can be real or complex.

2. Coherence Generation.
To calculate the degree of coherence between two co-registered complex images.

Coregistration

General
The coregistration function will co-register one (slave) image, or a set of (slave) images with respect to one (master) image. The coregistration function is fully automatic, in the sense that it does not require the user to manually select ‘tie points’ from the master and slave images.
The coregistration is performed in 2 stages for real images and in 3 stages for complex images.
(1) A coarse registration stage is achieved using the satellite orbit parameters.
(2) A fine registration stage is achieved using a cross correlation operation on a series of ‘cells’ defined across the images.
(3) The additional third registration stage for complex images is achieved by the maximisation of the complex coherence between the images.
Input Images
The input images for the coregistration function can be complex like SLC or SLCI (but the RAW is not permitted) or real like the PRI, GEC or GTC. The coregistration function requires that all of the input images belong to the same type (i.e. are all complex or all real) and have the same projection system (all slant range or all ground range projected or all geocoded). It is therefore not possible to mix product types, although, the GEC format can been registered with the GTC because they are both presented in the east, north projection.
Ground Control Points
The various coregistration stages operate on portions within the images defined by a series of ground control points. The generation of the ground control points (on the master image) is controlled by two user defined numbers which specify how many ground control points shall be taken in the row direction and in the column direction (the total ground control points number being the product of the two sizes).
For the coarse registration stage the system finds the exact correspondence of the ground control points on the various slave images using the geometric acquisition model (orbit, timing).
The fine registration stage (and the coherence maximisation for complex images) make use of image ‘cells’, which are image windows, with sizes determined by the user, based at the ground control points.
The speed and precision of the fine registration and coherence maximisation stages is strongly dependent on the size of the cross correlation ‘cells’. In general, the greater is the size of the cells, the more accurate is the coregistration but the more time consuming is the operation. Moreover, if very large cells are used it is possible to lose small local changes in the position of the ground control points, obtaining a poor registered set of images.
The accuracy of the fine registration operation is also determined by the setting of the interpolation factor that is used in the cross correlation of the cells. This factor effectively determines the step size that is used to move the slave cells with respect to the master cells during the cross-correlation process.
For complex data only, the registration is refined by shifting the master and slave cells with respect to each other until their coherence reaches a maximum. The accuracy and speed of this process is related to the coherence cell window sizes and the precision factor for the maximum coherence search.
Specified Ground Control Points
In some cases it can be useful for the user to select the ground control points on the master image on specified positions. This is possible by providing a text file with the required ground control point coordinates on the master image, in row, col coordinate system. (Obviously, there is no need to specify the slave positions of the ground control points as these are computed by the system.) This option is useful for registering images which contain large incoherent zones (like sea areas). If the ground control points are not specified directly, then they will be uniformly distributed in the image and it may be the case that only a few of them are placed in terrain zones. In contrast, by using the GCP (ground control point) file to specify ground control points, it is possible to avoid this behavior, and concentrate them in the coherent regions. Clearly the disadvantage of the ‘Specified Ground Control Point’ method is that it is necessary for the user to do some extra work to obtain the full resolution coordinates of the required ground control points. This can be done, for example, using the quick look function with the grid in row, col coordinates.
Another important use of this feature is the registration of ascending and descending pass images. In this case, the various image features are observed from different angles and are therefore very difficult to correlate. One solution for this problem is to select as ground control points only the high reflecting scatters (or similar regions). This can be done as explained before, with the quick-look with the grid in row, col coordinates.
The following example shows the format of the Specified Ground Control Point file, simply constituted by the various ground control points positions, expressed in the master full resolution row, col coordinate system:

```
100 200
244 772
844 902
1200 30
2309 4445
3531 552
7983 3444
```

Excluding the Worst Ground Control Points
It is possible to select only the best ground control points by excluding the ground control points that cannot be properly fitted by a polynomial warping function. This is achieved by making an initial generation of the warp function using all of the ground control points and then excluding the worst ground control points (i.e. those which generate the higher residuals). In this way only a set of ground control points which are compatible with the polynomial warp function are selected. This function is controlled by one parameter (Editing RMS) which defines the threshold for deciding if a ground control point shall be discarded or not.

The Warp Function
Once the valid positions of the ground control points in the slave image(s) are known, a function is computed using a least square method, which maps the ground control points in the master image onto the ground control points in the slave image. This function (usually know as the warp function) is used to perform the coregistration.
The warp function is a polynomial in the row, col coordinates of a selectable degree ranging from 1 (which corresponds to a simple transformation which includes only translation, rotation, scaling and skew of the two images) up to 3 (which is a rather complex transformation with no simple geometric explanation). The forms of the warp functions for the four different degrees (1, 1.5, 2 or 3) are described in “Warp Evaluation” chapter of the Algorithmic Specification Document [A3].
Important: As a simple rule, when the input images does not suffer from a high level of distortion try first the degree 1 (or 1.5). This permits the evaluation of a smooth warp, which is enough for the majority of cases (interferometric SLC couples included).
Use the higher degrees only when a very good coregistration accuracy is wanted, but be aware that high degree polynomials can introduce large distortions in image regions containing only a few ground control points. Therefore, in general, if the higher degree polynomials are used, large numbers of ground control points will be required, especially in areas that are of particular interest to the user.

Ground Control Point Residuals
A parameter is used to enable the generation of a text file which contain the residuals (the errors introduced by the warping function) for each ground control point. In this file are also contained the warp coefficients. The following example shows the format of this file, which can be useful to know the loss of registration accuracy introduced by the warp function:

```
### Slave Number =  1

 Transformation Degree = 1.000000
 Master to Slave Warp Matrix:
    62.34320     0.99937     0.00158
    -3.89881    -0.00005     0.99821
 Slave to Master Warp Matrix:
   -62.38784     1.00063    -0.00158
     3.90274     0.00005     1.00180

    | GCP master row | GCP master col | GCP slave row  | GCP slave col  |  Row Residuals |  Col Residuals |     Points RMS |
 -----------------------------------------------------------------------------------------------------------------------
  1 |       167.000 |       200.000 |       229.811 |       195.787 |       -0.256 |       -0.053 |        0.262
  2 |       333.000 |       200.000 |       395.066 |       195.646 |        0.384 |        0.079 |        0.392
  3 |       667.000 |       200.000 |       729.368 |       195.734 |       -0.127 |       -0.026 |        0.130
  4 |       667.000 |       400.000 |       729.556 |       395.349 |        0.000 |        0.000 |        0.000
  5 |       833.000 |       200.000 |       894.843 |       195.818 |        0.294 |       -0.118 |        0.316

 mean residuals in the rows    =     0.059
 mean residuals in the columns =    -0.024

 RMS of the residuals in the rows    =     0.244
 RMS of the residuals in the columns =     0.065

 Total mean =     0.018
 Total RMS  =     0.154


 ### Slave Number =  2

 Transformation Degree = 1.000000
 Master to Slave Warp Matrix:
    -0.00005     1.00000     0.00000
     0.00001     0.00000     1.00000
 Slave to Master Warp Matrix:
     0.00005     1.00000     0.00000
    -0.00001     0.00000     1.00000

    | GCP master row | GCP master col | GCP slave row  | GCP slave col  |  Row Residuals |  Col Residuals |     Points RMS |
 -----------------------------------------------------------------------------------------------------------------------
  1 |       167.000 |       200.000 |       167.000 |       200.000 |        0.000 |        0.000 |        0.000
  2 |       167.000 |       400.000 |       167.000 |       400.000 |        0.000 |        0.000 |        0.000
  3 |       167.000 |       600.000 |       167.000 |       600.000 |        0.000 |        0.000 |        0.000
  4 |       167.000 |       800.000 |       167.000 |       800.000 |        0.000 |        0.000 |        0.000
  5 |       333.000 |       200.000 |       333.000 |       200.000 |        0.000 |        0.000 |        0.000
  6 |       333.000 |       400.000 |       333.000 |       400.000 |        0.000 |        0.000 |        0.000
  7 |       333.000 |       600.000 |       333.000 |       600.000 |        0.000 |        0.000 |        0.000
  8 |       333.000 |       800.000 |       333.000 |       800.000 |        0.000 |        0.000 |        0.000
  9 |       500.000 |       200.000 |       500.000 |       200.000 |        0.000 |        0.000 |        0.000
 10 |       500.000 |       400.000 |       500.000 |       400.000 |        0.000 |        0.000 |        0.000
 11 |       500.000 |       600.000 |       500.000 |       600.000 |        0.000 |        0.000 |        0.000
 12 |       500.000 |       800.000 |       500.000 |       800.000 |        0.000 |        0.000 |        0.000
 13 |       667.000 |       200.000 |       667.000 |       200.000 |        0.000 |        0.000 |        0.000
 14 |       667.000 |       400.000 |       667.000 |       400.000 |        0.000 |        0.000 |        0.000
 15 |       667.000 |       600.000 |       667.000 |       600.000 |        0.000 |        0.000 |        0.000
 16 |       667.000 |       800.000 |       667.000 |       800.000 |        0.000 |        0.000 |        0.000
 17 |       833.000 |       200.000 |       833.000 |       200.000 |        0.000 |        0.000 |        0.000
 18 |       833.000 |       400.000 |       833.000 |       400.000 |        0.000 |        0.000 |        0.000
 19 |       833.000 |       600.000 |       833.000 |       600.000 |        0.000 |        0.000 |        0.000
 20 |       833.000 |       800.000 |       833.000 |       800.000 |        0.000 |        0.000 |        0.000

 mean residuals in the rows    =     0.000
 mean residuals in the columns =     0.000

 RMS of the residuals in the rows    =     0.000
 RMS of the residuals in the columns =     0.000

 Total mean =     0.000
 Total RMS  =     0.000
```

Interpolation
After the warp function has been computed, the next step in the coregistration process is the interpolation of the slave image(s) pixels onto the master image pixels. The interpolating function is configurable and many types are allowed.
The fastest (and least accurate) interpolators are the nearest neighbor, bilinear or constant shift, while a most accurate one is the cubic convolution. The most precise is the sinc, which uses a configurable kernel size to obtain the interpolated value. It is also possible to mix two different types of interpolation; one method for the row direction and another for the column direction (this facility may be useful depending on the distortions suffered by the SAR images, for example, for interferometric pairs, the distortion in the column direction is often very low compared to the row direction).
The following interpolators can be used:

- **nearest neighbor**, takes the pixel nearest to the position determined by the warp function; has an intrinsic accuracy of 0.5 pixel but is very fast
- **bilinear**, uses three linear interpolations of the four pixel values around the position determined by the warp function
- **cubic convolution**, uses five interpolations with a four coefficient cubic convolution kernel applied to the sixteen pixels around the position determined by the warp function
- **sinc**, which is the best (and slowest) interpolator, uses a sinc kernel of a selectable size **N**, applied **N+1** times to the **N** by **N** pixels around the position determined by the warp function
- **constant shift**, in which each image block which has to be interpolated, is shifted by a unique amount (depending by the interpolation grid), by means of a FFT operation
- **sinc along rows and constant shift along columns**, in which the previous two different interpolators are used along rows and along columns
- **cubic convolution along rows and constant shift along columns**, in which the previous two different interpolators are used along rows and along columns

Overlap Selection
The last parameter to configure is associated with the selection of the image zone portion which shall be co-registered.
Such zone can be selected with the following criteria:

- **AOI**, in which the image portion to be registered is directly specified by the user. Any kind of AOI can be used except the example image mode.
- **minimum overlap**, where the image portion to be registered contains the pixels which are present both in the master and in all the slaves (this portion corresponds to the overlap zone between all the images of the input stack) see “Fig. 36 - The minimum overlap scheme”
- **maximum overlap**, where the image portion to be registered contains the pixels which are present in at least one image (this portion correspond to common and non common zone between all the images of the input stack) see “Fig. 37 - The maximum overlap scheme”
- **master overlap**, where the image portion to be registered contains the pixels belonging to the master image, see “Fig. 38 - The master overlap scheme”

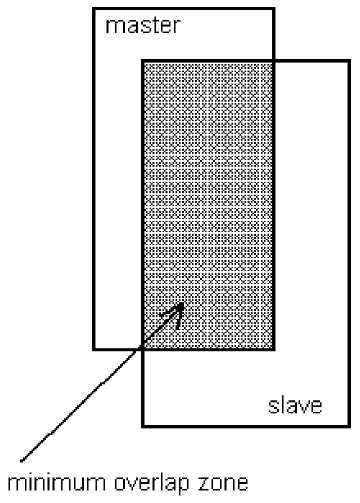


Fig. 36 - The minimum overlap scheme

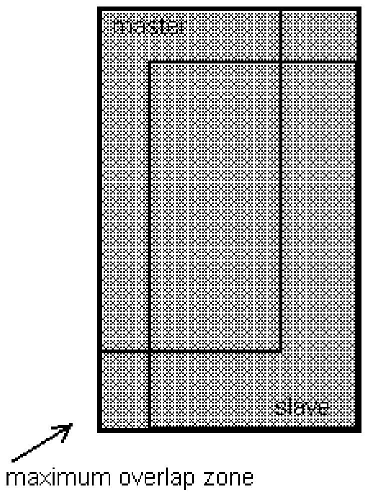


Fig. 37 - The maximum overlap scheme

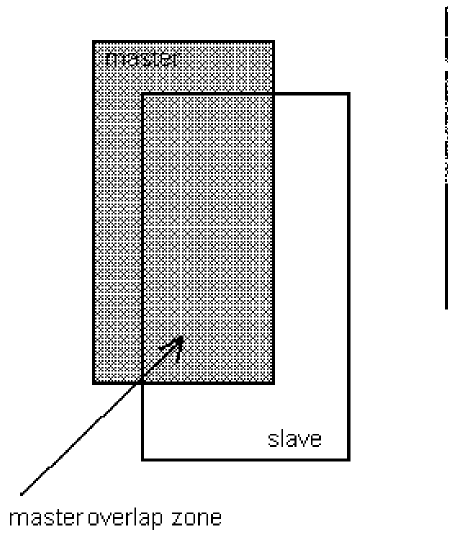


Fig. 38 - The master overlap scheme

Baseline Calculation
For real or complex images, the baseline between the two satellites can be calculated using the orbital and the warp function information. The baseline is evaluated at the scene centre (both the normal and parallel components) in the same coordinate system in which the orbit is given for each slave image. It is possible for the user to specify the name of the text file which will contain these values.
The following example shows the format of the baseline file for a single slave case:

```
### Slave number 1

 Baseline Cartesian Components in meters:

       X              Y              Z
 --------------------------------------------
   -30.535631      -60.703657      19.806008

 Baseline Components in the Sat - Target Plane in meters:

          Normal          Parallel
 --------------------------------------------
        62.978227        32.301387
```

Example .ini files

The following “.INI” files are examples for the coregistration tool. Two cases are shown, the most basic, with the minimum set of parameters and a more complicated one, with the customisation of the main configuration parameters.

```
[IMAGE CO-REGISTRATION]
Input Dir = "./"
Output Dir = "./"
Input Images = "slc_master.XTt", "slc_slave.XTt"
Output Images = "master", "slave"
```

```
[IMAGE CO-REGISTRATION]
Input Dir = "./"
Output Dir = "./"
Input Images = "pri_master.XTt", "pri_slave1.XTt", "pri_slave2.XTt"
Output Images = "master", "slave1", "slave2"
Baseline File Name = "basel.txt"
Top Left Corner = 100, 200
Bottom Right Corner = 1500, 3000
GCPs Numbers = 5, 3
Coarse Reg Window Sizes = 101, 101
Coarse Reg Interp Factors = 3., 3.
Transformation Degree = 2
Overlapping Mode = "MIN"
Interpolation Mode = "SINC"
```

Coregistration Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Images`	the name of the input images in internal format to be coregistered `Input Images = "mas.XTt", "sla1.XTt", "sla2.XTt", "sla3.XTt"`	mandatory parameter; a maximum of 10 images can be used
AOI specification	see Appendix 4 (no example image can be used) Note that the AOI specification prevents the use of the minimum, maximum and master overlap schemes	optional parameter; if not present, the overlap scheme parameter is used
`Output Images`	the name of the output images in internal format containing the coregistered image (an extension "`CRf or CRc`" is automatically added by the system) `Output Images = "masr", "sla1r", "sla2r", "sla3r"`	mandatory parameter
`Baseline File Name`	this parameter is used to specify the text file which will contain the normal and parallel components of the baseline, evaluated for each slave image `Baseline File Name = "basel_values.txt"`	optional parameter; if not present the name `baseline.txt` is assumed
`GCPs Numbers`	the number of GCPs along the row and along the column direction `GCPs Numbers = 7, 5`	optional parameter; if absent a value of `10, 10` is assumed; these values are however not used when the `GCPs File Name` parameter is present
`GCPs File Name`	this parameter is used to specify the text file which will contain the GCPs positions on the master image, expressed in row,col full resolution coordinates `GCPs File Name = "GCPFile.dat"`	optional parameter; if not present the GCPs are generated using the `GCPs Numbers` parameter
Editing RMS	this parameter is used to select the threshold for excluding the GCPs which originates the high residual errors Editing RMS = 1.0	optional parameter; if not present the value 1.0 is assumed
`Coarse Reg Window Sizes`	the row,col sizes of the cell used for the coarse GCPs position determination, using the cell cross correlation `Coarse Reg Window Sizes = 51, 51`	optional parameter; if absent a size of `51, 51` is assumed
`Coarse Reg Interp Factors`	the row,col interpolation factors of the cross correlation cell used for the coarse GCPs position determination `Coarse Reg Interp Factors = 3.5, 3.5`	optional parameter; if absent a size of `1.0, 1.0` is assumed
Coarse Reg Tolerance	the threshold used to check that the GCP position on slave images (computed with the cross correlation) is stable; the value corresponds to the maximum allowed difference (in pixel units) between two measures of the GCP position executed slightly changing the cell (the GCP position shall not depend too much on the particular cell being used) Coarse Reg Tolerance = 1.0	optional parameter; if absent 1.1 is assumed

Real Image Fine Reg	this parameter permits to enable the fine registration step (always done for complex images) also for real images Real Image Fine Reg = 'Y'	optional parameter; if absent no fine registration is done for real images is assumed
`Fine Reg Window Sizes`	the row,col sizes of the cell used for the fine GCPs position determination, using the coherence maximization `Fine Reg Window Sizes = 51, 51`	optional parameter; if absent a size of `51, 51` is assumed
`Coherence Window Size`	the row,col sizes of the small square moving window used for the coherence evaluation in the fine GCPs position determination `Coherence Window Size = 5`	optional parameter; if absent a size of `3` is assumed
`Coherence Func Tolerance`	the first stop criteria for the iterative search in coherence maximization (when the coherence changes are below the tolerance, the search stops) `Coherence Func Tolerance = 1.e-6`	optional parameter; if absent a size of `1.e-6` is assumed
`Coherence Value Tolerance`	the second stop criteria for the iterative search in coherence maximization (when the shifts made on the slave cell are below the tolerance, the search stops) `Coherence Value Tolerance = 1.e-3`	optional parameter; if absent a size of `1.e-3` is assumed
`Transformation Degree`	the degree of the warp transformation selected between: • `1` • `1.5` • `2` • `3` `Transformation Degree = 1.5`	optional parameter; if absent a degree of `1.5` is assumed
`Interp Window Sizes`	the size of the processing image block in which the master image is subdivided; this parameters affects the accuracy only when the interpolator is set to `CONSTANT SHIFT` (in one or both the directions); it affects however the speed (it is better to use large blocks) `Interp Window Sizes = 150, 150`	optional parameter; if absent a size of 512, 512 pixels is assumed
`Interpolation Mode`	the type of interpolation method selected between: • `NEAREST NEIGHBOUR` • `BILINEAR` • `CONSTANT SHIFT` • CUBIC CONVOLUTION • `SINC` • `CONSTANT SHIFT CUBIC CONV` • `CONSTANT SHIFT SINC` `Interpolation Mode = "CUBIC CONVOLUTION"`	optional parameter; if absent the `CUBIC CONVOLUTION` mode is assumed

`Interpolation Inverse Precision`	the length of lookup tables used to speed up the sinc interpolation; a high value guarantee a good accuracy (comparable with the reciprocal of this size, i.e. 1/1000 of pixel) at expense of a limited memory usage `Interpolation Inverse Precision = 1000`	optional parameter; if absent a size of `1000` entries is assumed
`Sinc Width`	the size of the sinc kernel; this value affects the accuracy of the interpolation at expense of the processing time `Sinc Width = 7`	optional parameter; if absent a kernel size of `7` coefficients is assumed
`Cubic Convolution Coefficient`	the coefficient which modify the behavior of the cubic convolution interpolator; the IDL cubic interpolation uses a coefficient equal to `-1` while ERDAS suggest `+0.5` `Cubic Convolution Coefficient = -1.0`	optional parameter; if absent a value equal to `-1` is assumed
`Overlapping Mode`	the kind of overlapping scheme selected between: • `MIN` • `MAX` • `MASTER` `Overlapping Mode = "MASTER"`	optional parameter; if absent the `MASTER` mode is assumed
`Residual File Name`	the name of the file which details the various residuals (errors introduced by the warp function) on each GCP, along with the warp coefficients `Residual File Name = "residual"`	optional parameter; if absent no residual file is generated

Coherence generation

This function is used to generate the coherence image between two co-registered complex SAR products. The coherence is generated in a window of a user-defined size, which moves in with a step size of 1 pixel across the co-registered images.
The only images that can be used as input are complex images in internal SAR Toolbox format having a floating point pixel representation. In other words, images can not be input that have simply been extracted from a SLC SAR product, in which the pixel has a complex integer format.
The coherence generation function product produces an output image with the same size as the input couple. Note that there will be an 'edge effect' caused by the size of the moving window in which the coherence is calculated. This effect will cause a ring of pixels (with a depth equal to half the window size) to be set to zero at the edges of the output image.The following ".INI" files is an example for the coherence image generation.

Example .ini file

```
[COHERENCE IMAGE GENERATION]
Input Dir = "./"
Output Dir = "./"
Input Images = "slc_master.CRc", "slc_slave.CRc"
Output Image = "cohe"
```

Coherence Generation Summary Table

Parameter	Description **Example**	Comment
`Input Images`	the name of the input image couple in internal format from which the coherence is generated `Input Images = "mas.CRc", "sla.CRc"`	mandatory parameter
AOI specification	see Appendix 4 (no example image neither polygonal specification can be used)	optional parameter; if not present, the entire images are assumed
Window Sizes	the size of the moving window used to compute the coherence; a couple of integer numbers comma separated, the first one referring to the number of rows, the second to the number of columns Window Sizes = 5, 5	mandatory parameter
`Output Image`	the name of the output image in internal format containing the coherence (an extension "`CHf`" is automatically added by the system) `Output Image = "cohe"`	mandatory parameter

9 Speckle filtering tool

The speckle filter tool removes speckle noise from intensity images using the ‘Gamma MAP’ algorithm. The speckle filter tool operates on real intensity images in the internal SAR Toolbox format.
If the input data is from a PRI image, this will initially be in amplitude form and will therefore first need to be converted to an intensity image (i.e. using the Amplitude to Power function).
Or if the data is from a SLC image, the data will first need to be converted to a real form (i.e. Modulus extraction) followed by a conversion to intensity (i.e. the Amplitude to Power function).
Of course, if required, the output from the Speckle Filter function may be converted from intensity to amplitude form by using the Power to Amplitude function.
The Speckle Filter makes use of a range of different masks that are applied in a moving window that covers the input image. For each position of the window the masks are used to determine the structure of the dominate feature within the window. Once this structure has been discovered, the filter may either preserve the pixel vale at the centre of the window, replace it with the mean of the window values (in the case for homogeneous regions, or use the GAMMA - GAMMA MAP filter to evaluate the centre pixel value.
Further details about the masks used in the Speckle Filter algorithms are given after the Speckle Filter Summary Table.

Further characteristics of the speckle filter tool are:

- uses only the rectangular AOI specification with corners expressed in row, col system
- uses (for the standard masks) odd filter sizes ranging from 3 by 3 to 31 by 31 with a correspondingly number of directions from 4 to 60 (the upper kernel size limitation is only imposed by the automatic computation of the thresholds; if these values are inserted from the user a greater kernel can be used)
- used predefined masks evaluating in an automatic way the edge and line thresholds
- can also apply user defined masks, once the edge and line thresholds are specified by the user
- updates in the output image, the following annotations:
    - ⇒ the new values of the coordinates in lat, lon of the four corners and of the center point
    - ⇒ the new entry in the processing history
- is interfaced with the IDL system at linkimage level, i.e. the SAR Toolbox functions when activated became part of the system and can be called as any other IDL function
- the speckle filtering function is interfaced with ERMAPPER using the “filter” feature

Example .ini files

The following “.INI” file is an example for the speckle filtering of a PRI image portion with standard masks:

```
[SPECKLE FILTER]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.APf"
Top Left Corner = 50, 50
Bottom Right Corner = 500, 500
Window Sizes = 11, 11
PFA = 10.0
Scatter Threshold = 0.57
Output Image = "specklefiltered_img"
```

The following file does the same operation with user mask

```
[SPECKLE FILTER]
Input Dir = "./"
Output Dir = "./"
Input Image = "input.APf"
Top Left Corner = 50, 50
Bottom Right Corner = 500, 500
Window Sizes = 11, 11
PFA = 10.0
Scatter Threshold = 0.57
Mask File = "user_mask.ker"
Number of Look = 3.0
Edge Threshold = 0.82
Line Threshold = 0.87
Output Image = "specklefiltered_img"
```

Speckle Filter Summary Table

Parameter	**Description** **Example**	**Comment**
`Input Image`	the name of the real input image in internal format, containing intensity (i.e. power) data; this image can come from a PRI or a SLC data but in the latter case a modulus extraction shall be performed before the power conversion `Input Image = "t1_priimage.APf"`	mandatory parameter
AOI specification	only the Rectangular AOI specifications are permitted (see Appendix 4)	optional parameter; if not present, the entire input images are assumed
`Window Sizes`	the size of the speckle filter in pixel units (row,col ordering) `Window Sizes = 11, 11`	mandatory parameter
`PFA`	the Probability of False Alarm in the various regions recognition, expressed in percentage units (lower is this value, less filtered is the output image, due to the fact that we are recognizing the various regions under a very strict criterion) `PFA = 10.0`	mandatory parameter
`Scatter Threshold`	the threshold used for the scatter detection (higher is this value, less preserved are the scatters in the output image, due to the fact that we are recognizing the recognizing regions under a very strict criterion) `Scatter Threshold = 0.57`	mandatory parameter
`Output Image`	the name of the output image in internal format containing the speckle filtered intensity image (an extension "`SFf`" is automatically added by the system) `Output Image = "specklefiltered_img"`	mandatory parameter
`Mask File`	the filename containing filtering masks defined by the user; the user masks are used instead of the standard when this parameter is present `Mask File = "usermask.ker"`	mandatory parameter for the filtering with user masks
`Number of Look`	the number of looks of the input image; when this parameter is absent from the .INI file, then the value stored in the image annotations is used; due to the change of the image statistics after a image filtering operation, the output image has this value set to 0 in the annotations, thus requiring the user to set a right value (e.g. obtained from measures on the filtered image) `Number of Look = 3.0`	optional parameter for images directly coming from the extraction tool mandatory parameter for input images which have been previously filtered
`Edge Threshold`	the value of the threshold for the ratio detector applied on the edge regions of user masks; this value should be computed using the relation showed in Speckle Filter chapter of the document ; for standard masks this value is automatically computed by the system but can be however overridden with this .INI line `Edge Threshold = 0.82`	optional parameter for standard masks mandatory parameter for user masks

`Line Threshold`	the value of the threshold for the ratio detector applied on the edge regions of user masks; this value should be computed using the relation showed in Speckle Filter chapter of the document ; for standard masks this value is automatically computed by the system but can be however overridden with this .INI line `Line Threshold = 0.87`	optional parameter for standard masks mandatory parameter for user masks

The filtering masks are contained in ASCII files and their structure is governed by the following rules:

- all the masks for the various regions (edge, line, scatter) filtering are kept in a unique ASCIII file
- all the masks shall have the same size, which defines the filter size
- each row shall be terminated with a newline (return) character
- the first row of each mask shall indicate
  - ⇒ the mask type, chosen between “edgeline” and “scatter”
  - ⇒ the mask identification, as a progressive number between 1 and the number of possible masks of a given type
- the symbols used for the pixel description shall be separated by **one space character**
- the symbol used to describe a left edge region is the “1”
- the symbol used to describe a right edge region is the “4”
- the symbol used to describe a line region is the “.”
- the symbol used to describe a left buffer region is the “2”
- the symbol used to describe a right buffer region is the “3”

The following listing shows the standard mask file of size 11:

```
edgeline 1
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
edgeline 2
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
3 3 4 4 4 4 4 4 4 4 4
. . 3 3 4 4 4 4 4 4 4
2 2 . . 3 3 3 4 4 4 4
1 1 2 2 . . . 3 3 4 4
1 1 1 1 2 2 2 . . 3 3
1 1 1 1 1 1 1 2 2 . .
1 1 1 1 1 1 1 1 1 2 2
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
edgeline 3
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 2 2
1 1 1 1 1 1 1 2 2 . .
1 1 1 1 2 2 2 . . 3 3
1 1 2 2 . . . 3 3 4 4
2 2 . . 3 3 3 4 4 4 4
. . 3 3 4 4 4 4 4 4 4
3 3 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
edgeline 4
. 3 4 4 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
1 2 . 3 4 4 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 1 1 2 . 3 4
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 1 1 2 .
edgeline 5
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2
. . . . . . . . . . .
3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
edgeline 6
1 1 1 1 1 1 1 1 1 2 .
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 2 . 3 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 2 . 3 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
. 3 4 4 4 4 4 4 4 4 4
edgeline 7
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
edgeline 8
1 1 2 . 3 4 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 1 2 . 3 4 4
```

```
scatter 1
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . X . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
. . . . . . . . . . .
```

The user masks can be built in an analogous way. The most important thing to remember when building user mask are:

- for user masks, the edge and line thresholds cannot be computed by the system (refer to the Speckle Filter chapter of the document [A3] for details) and shall be given by the user (the mathematical computations are however quite complex and also use empirical formulas); in the Speckle Filter chapter of the document [A3] some values are given for certain filter sizes and PFA values
- the various regions (e.g. the line or the edge) shall always be centred on the central mask pixel

The user masks can be useful if it is required that the speckle filter recognize special image features, like the curved line in the following mask:

```
edgeline 1
2 . 3 4 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
2 . 3 3 4 4 4 4 4 4 4
2 2 . . 3 3 3 4 4 4 4
1 1 2 2 . . . 3 3 4 4
1 1 1 1 2 2 2 . . 3 3
1 1 1 1 1 1 1 2 2 . 3
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 1 2 . 3
```

10 Calibration tools

The tools included in this section are:

1. Backscattering Image Generation.

To convert a power image into backscatter image, i.e. an image that is dependent only on the backscattering properties of the Earth’s surface and independent from the satellite and radar parameters.

2. ADC Compensation Image Generation.

This function corrects a power image for the ADC saturation phenomenon. The output may then be used as input in the Backscattering Image Generation function.

3. Gamma Image Generation.

This is to convert a backscatter image (i.e. output from Backscattering Image Generation) into a Gamma image. This is achieved by dividing the backscatter image by the cosine of the satellite incidence angle.

Backscattering image generation

The backscattering image generation is used to convert a power image into backscatter image. The following radiometric effects which could give a low precision backscattering image can be corrected with this tool:

- antenna pattern
- range spreading losses
- replica power variation
- ADC saturation effect

The first two affects the SLC data which originally comes from the SAR processor without these corrections while the remaining two alter the radiometry of both PRI or SLC products. The output image may be either in linear or dB scale.
When the ADC saturation correction is required, an ADC correction image shall also be provided as input. This correction image should be previously generated using the ADC Correction Image Generation function described in the following section. The Backscattering Image Generation function needs for each pixel the value of the ADC correction, so the ADC correction image must be computed on the same image portion which is used now to feed the backscattering data generation or in a larger portion. An error message is issued when this condition is not respected.
The steps then needed to obtain a fully precision backscattering image are then the following:
first, evaluate the ADC correction image starting from the input SAR product (or a portion of it)
then, apply the calibration constant, the ADC and some more corrections to the input SAR product (or a portion of it)
To make possible the Antenna Pattern correction application or removal, two Antenna Pattern Files shall be present. These files, which contain the two nominal (ERS1 or ERS2) patterns or even a user defined pattern, must be initially created using the Support Data Ingestion which, as described in the related section, transforms an ASCII file containing a lookup table (the antenna pattern in the source ESA format) in its STB internal representation.

Example .ini files

The following ".INI" file is the simpler example for a backscattering image generation from a PRI power image, without any correction for the replica power variation and ADC saturation:

```
[IMAGE BACKSCATTERING]
Input Dir = "./"
Output Dir = "./"
Input Image = "pri.APf"
Calibration Constant Correction = "APPLY"
Output Image Scale = "DB"
Output Image = "pri_s0"
```

The following ".INI" file is an example for the fully compensated backscattering image generation from a PRI power image coming from a PAF which annotates the replica power value (in the PRI the antenna pattern and the range spreading loss are already compensated during the SAR processing):

```
[IMAGE BACKSCATTERING]
Input Dir = "./"
Output Dir = "./"
Input Image = "pri.APf"
Replica Power Correction = "APPLY"
Reference Replica Power = 205229.0, 156000.0
ADC Saturation Correction = "APPLY"
ADC Saturation Correction File = "pri_adc.ADf"
Calibration Constant Correction = "APPLY"
Output Image Scale = "DB"
Output Image = "pri_s0"
```

The following “.INI” file is an example for the fully compensated backscattering image generation from a SLC power image coming from a PAF which annotates the chirp density value (in the SLC the antenna pattern and the range spreading loss are not corrected during the SAR processing and shall be compensated here):

```
[IMAGE BACKSCATTERING]
Input Dir = "./"
Output Dir = "./"
Input Image = "slc.APf"
Antenna Pattern Correction = "APPLY"
Range Spreading Loss Correction = "APPLY"
Replica Power Correction = "APPLY"
Reference Chirp Average Density = 267200, 201060
ADC Saturation Correction = "APPLY"
ADC Saturation Correction File = "slc_adc.ADf"
Calibration Constant Correction = "APPLY"
Output Image Scale = "DB"
Output Image = "slc_s0"
```

Backscattering Image Generation Summary Table

Parameter	Description Example	Comment
Input Image	the name of the input image in internal format containing intensity data, from which the backscattering image is generated Input Image = "pri.APf"	mandatory parameter
AOI specification	see Appendix 4 (no example image can be used)	optional parameter; if not present, the entire image is assumed
Antenna Pattern Correction	the flag indicating if the antenna pattern compensation factor shall be applied (APPLY) or removed (REMOVE) from the image; if this parameter is omitted, this correction is not considered at all (neither applied or removed) Antenna Pattern Correction = "APPLY"	optional parameter; if not present, no correction is done
Antenna Pattern File	the file name of the internal format data containing the antenna pattern Antenna Pattern File = "ers2.e6"	mandatory parameter if Antenna Pattern Correction is set to APPLY
Range Spreading Loss Correction	the flag indicating if the range spreading loss compensation factor shall be applied (APPLY) or removed (REMOVE) from the image; if this parameter is omitted, this correction is not considered at all (neither applied or removed) Range Spreading Loss Correction = "APPLY"	optional parameter; if not present, no correction is done
Replica Power Correction	the flag indicating if the replica power variation compensation factor shall be applied (APPLY) or removed (REMOVE) from the image; if this parameter is omitted, this correction is not considered at all (neither applied or removed) Replica Power Correction = "APPLY"	optional parameter; if not present, no correction is done
Reference Replica Power	the reference value of the replica power (in linear scale) for ERS1 and ERS2 Reference Replica Power = 205229.0, 156000.0	mandatory parameter if the following conditions are all true: Replica Power Correction is set to APPLY the value of the replica power is found in the image
Reference Chirp Average Density	the reference value of the chirp average density (in linear scale) for ERS1 and ERS2 Reference Chirp Average Density = 267200, 201060	mandatory parameter if the following conditions are all true: Replica Power Correction is set to APPLY no replica power value is found in the image (in this case the value of the Reference Chirp Average Density is searched instead)

ADC Saturation Correction	the flag indicating if the ADC saturation compensation factor shall be applied (APPLY); if this parameter is omitted, this correction is not considered at all; this correction cannot be removed from the image ADC Saturation Correction = "APPLY"	optional parameter; if not present, no ADC correction is done
ADC Saturation Correction File	the internal format data containing the ADC saturation correction image, in input to the backscattering function ADC Saturation Correction File = "adc.ADf"	mandatory parameter if the ADC correction is required
Calibration Constant Correction	the flag indicating if a backscattering data shall be obtained from the input power image (APPLY) or if the reverse transformation from backscattering to original power image shall be instead done (REMOVE); if this parameter is omitted, this transformation is not done at all (neither in one direction or in the other); this feature permits to apply to the input image one or many correction factors but without transforming the data into a backscattering image Calibration Constant Correction = "APPLY"	optional parameter; if not present, no backscattering image is obtained in output
Calibration Constant	the user value of the calibration constant; if missing the value contained in the image annotations is used Calibration Constant = 1000000.0	mandatory parameter if the following conditions are all true: the calibration constant application or removal is required a user value of the calibration constant shall be used (instead the one contained in the SAR image annotations)
Output Image Scale	the type of output image scale chosen between: LINEAR DB Do not use the dB scale output if a further step of averaging is foreseen. Output Image Scale = "DB"	optional parameter; if missing a LINEAR scale is assumed
Output Image	the name of the output image in internal format containing the backscattering data (an extension “BSf” is automatically added by the system) Output Image = "backscatt"	mandatory parameter

ADC compensation image generation

This function permits to compute the ADC compensation image which will be used in the Backscattering Image Generation function, to correct for the ADC saturation phenomenon. This effect (present on the ERS images but particularly relevant only for the ERS1 sensor) can alter the derived backscattering values on high reflectivity zones. The ADC correction image shall be evaluated on the same power image that feeds the backscattering generation. No AOI can be inserted in the ADC computing, but this function can work on image portions. The important thing to remember is that the image in input to the ADC correction shall be or the same, or shall contain the image in input, to the backscattering generation. The ADC image generation is mainly based on two filters with averaging kernels; the first, called RMS averaging, is used just to reduce the computation load while the second, called smoothing, uses certain sizes depending on the length of the functions used during the SAR processing (which change from the SLC and the PRI products). All these sizes are user selectable, even if at least the smoothing parameters should be left to the preset values. During the ADC compensation image generation some radiometric corrections applied on the input image shall be removed so the related parameters (used during the backscattering image generation but to apply the radiometric correction) are needed. Some care must be taken avoiding changing these values between the ADC and the backscattering generation. As an example, if a user wants to select their own value for the calibration constant during the backscattering image generation, the same value shall be repeated here. The ADC correction generation uses a couple of tables called ADC lookup tables (one for ERS1 and the other for ERS2), kept in internal format. These two files (as done for the Antenna Pattern) must be initially created using the Support Data Ingestion which, as described in the related section, transforms an ASCII file containing a lookup table (the ADC correction lookup table in the source ESA format) in its STB internal representation.
No AOI can be used for the ADC correction image generation.

Example .ini file

```
[ADC COMPENSATION GENERATION]
Input Dir = "./"
Output Dir = "./"
Input Image = "pri.APf"
RMS Window Size = 8, 8
PRI Smoothing Window Size = 400, 1200
SLC Smoothing Window Size = 630, 1280
Reference Replica Power = 205229.0, 156000.0
Reference Chirp Average Density = 267200, 201060
Output Image = "pri_adc"
```

ADC compensation Image Generation Summary Table

Parameter	Description Example	Comment
Input Image	the name of the input image in internal format containing intensity data, from which the ADC compensation image is generated; this image should be the same or shall contain the image used for the backscattering Input Image = "pri.APf"	mandatory parameter
Reference Replica Power	the reference value of the replica power (in linear scale) for ERS1 and ERS2 Reference Replica Power = 205229.0, 156000.0	mandatory parameter when the value of the replica power (instead the chirp average density) is found in the image
Reference Chirp Average Density	the reference value of the chirp average density (in linear scale) for ERS1 and ERS2 Reference Chirp Average Density = 267200, 201060	mandatory parameter when the value of the chirp average density (instead the replica power) is found in the image
Calibration Constant	the user value of the calibration constant; if missing the value contained in the image annotations is used; if used, this parameter shall be the same adopted for the backscattering image generation Calibration Constant = 9500000.0	mandatory parameter if a user value of the calibration constant is adopted in the backscattering image generation
RMS Window Size	the two sizes used to shrink the input image and reduce the computation load of the ADC correction evaluation; as usual, the first refers to the row size the second to the column size RMS Window Size = 8, 8	mandatory parameter
PRI Smoothing Window Size SLC Smoothing Window Size	the sizes of the smoothing window (the first refers to the row size the second to the column size); due to the different values which shall be used for the SLC or the PRI data, and to avoid the need of changing this parameter every time the product changes, both values (for the SLC and the PRI) shall be inserted (even if only one is used at a time) PRI Smoothing Window Size = 400, 1200 SLC Smoothing Window Size = 630, 1280	mandatory parameter
Output Image	the name of the output image in internal format containing the backscattering data (an extension “ADf” is automatically added by the system) Output Image = "pri_adc "	mandatory parameter

Gamma Image Generation

The Gamma Image Generation function converts a backscatter image (i.e. output from Backscattering Image Generation) into a Gamma image. This is achieved by dividing the backscatter image by the cosine of the satellite incidence angle. For certain terrain types the application of this function will make the backscatter image independent of the satellite incidence angle. An Area of Interest (AOI) (all types except the example image) can be specified and the output scale can be set to dB.

Example .ini file

The following ".INI" file is the example for a Gamma image generation:

```
[IMAGE GAMMA]
Input Dir = "./"
Output Dir = "./"
Input Image = "sigma_0.BSf"
Output Image Scale = "DB"
Output Image = "gamma"
```

Gamma Image Generation Summary Table

Parameter	Description Example	Comment
Input Image	the name of the input image in internal format containing backscattering data Input Image = "sigma_0.APf"	mandatory parameter
AOI specification	see Appendix 4 (no example image can be used)	optional parameter; if not present, the entire image is assumed
Output Image Scale	the type of output image scale chosen between: LINEAR DB Do not use the dB scale output if a further step of averaging is foreseen. Output Image Scale = "DB"	optional parameter; if missing a LINEAR scale is assumed
Output Image	the name of the output image in internal format containing the backscattering data (an extension "GAf" is automatically added by the system) Output Image = "gamma"	mandatory parameter

Appendix 1: Example of the ASCII Header Analysis output file

An example of the ASCII Header Analysis output file is shown here. However, these files are very long and this listing it is only a subsection, which provides an example of its format.
The following information is given in the six columns:
parameter sequential number
name of the field as appears in the ESA format documentation
value of the parameter
units in which the parameter is expressed
internal field name, as appears in the parameter dump obtained with the data conversion tool
a remark

```
==========================================================================================================================
==============================
STB - Sar Tool Box - Telespazio / ESA - ANNOTATION LIST
==========================================================================================================================
==============================
Processing time............: 26-Jan-1998 15:57:52.000
Product type...............: PRI
Source.....................: ERS-2
Data format................: CEOS
Facility id................: DEP
Format descriptor record...: ./dat/pri2cdep
-----------------------------------------------------------------------------------------------------------------------
File name..................: VDF - VOLUME_DIRECTORY_FILE
Record name................: Volume Descriptor Record
 Pos Esa field name                    Value              Units        Tag                  Remark
-----------------------------------------------------------------------------------------------------------------------
   1 Record sequence number                1                   -            rec_seq_num            -
-----------------------------------------------------------------------------------------------------------------------
   2 1st record sub-type code            192                  -            rec_subtype_code_1       -
-----------------------------------------------------------------------------------------------------------------------
   3 Record type code                   192                  -            rec_type_code            -
-----------------------------------------------------------------------------------------------------------------------
   4 2nd record sub-type code             18                  -            rec_subtype_code_2       -
-----------------------------------------------------------------------------------------------------------------------
   5 3rd record sub-type code             18                  -            rec_subtype_code_3       -
-----------------------------------------------------------------------------------------------------------------------
   6 Length of this record              360                  -            rec_len                  -
-----------------------------------------------------------------------------------------------------------------------
   7 ASCII/EBCDIC Flag                     A                  -            ascii_ebcdic_flag        -
-----------------------------------------------------------------------------------------------------------------------
   8 Blanks                         -                  -            blanks                   -
-----------------------------------------------------------------------------------------------------------------------
   9 Format control document              CCB-CCT-0002            -            format_control_id        -
-----------------------------------------------------------------------------------------------------------------------
  10 Superstructure format control document     E                  -            ss_format_control        -
-----------------------------------------------------------------------------------------------------------------------
  11 Superstructure record format revision     A                  -            ss_format_revision       -
-----------------------------------------------------------------------------------------------------------------------
  12 Logical volume generating facility softw   ERS2-PRI-6.0            -            log_vol_id               -
     are release and revision level
-----------------------------------------------------------------------------------------------------------------------
  13 ID of physical volume containing this vo   343101                  -            phys_tape_id             -
     lume descriptor
-----------------------------------------------------------------------------------------------------------------------
  14 Logical volume identifier              0005244200004805        -            release_revision         -
-----------------------------------------------------------------------------------------------------------------------
  15 Volume set identifier                1996052910374928        -            vol_set_id               -
-----------------------------------------------------------------------------------------------------------------------
  16 Total number of physicals volumes in the   1                  -            nb_phys_vol              -
      logical volume
-----------------------------------------------------------------------------------------------------------------------
  17 Physical volume sequence number of the f   1                  -            phys_vol_seq_first       -
     irst tape within the logical volume
-----------------------------------------------------------------------------------------------------------------------
  18 Physical volume sequence number of the l   1                  -            phys_vol_seq_last        -
     ast tape within the logical volume
-----------------------------------------------------------------------------------------------------------------------
  19 Physical volume sequence number of curre   1                  -            phys_vol_seq_cur         -
     nt tape within the logical volume
-----------------------------------------------------------------------------------------------------------------------
  20 First referenced file number in this phy   1                  -            first_fl_num             -
     sical volume within the logical volume
-----------------------------------------------------------------------------------------------------------------------
  21 Logical volume number within volume set    1                  -            log_in_vol_set           -
-----------------------------------------------------------------------------------------------------------------------
  22 Logical volume number within physical vo   1                  -            log_in_phys_vol          -
     lume
-----------------------------------------------------------------------------------------------------------------------
  23 Logical volume creation date (YYYYMMDD)    19970312                -            creation_date            -
-----------------------------------------------------------------------------------------------------------------------
  24 Logical volume creation TIME (HHMMSSDD,    172631                  -            creation_time            -
     DD=deci-seconds)
-----------------------------------------------------------------------------------------------------------------------
  25 Logical volume genration country           GERMANY                 -            log_vol_gen_country      -
-----------------------------------------------------------------------------------------------------------------------
  26 Logical volume genration agency            ESA                  -            log_vol_gen_agency       -
-----------------------------------------------------------------------------------------------------------------------
  27 Logical volume genration facility          D-PAF                -            log_gen_facility         -
-----------------------------------------------------------------------------------------------------------------------
  28 Number of file pointer records in volume   2                  -            nb_fl_ptr_recs           -
      directory
-----------------------------------------------------------------------------------------------------------------------
  29 Number of records in volume directory      4                  -            nb_recs                  -
-----------------------------------------------------------------------------------------------------------------------
  30 Total number of logical volume set         1                  -            tot_nb_log_vol           -
-----------------------------------------------------------------------------------------------------------------------
  31 Volume descriptor spare segment (always    -                  -            vol_descr_spare          -
```

Pos	Esa field name	Value	Units	Tag	Remark
	blanks filled)				
32	Local use segment	-	-	local_use_segment	-

Record name................: Leader File Pointer Record

Pos	Esa field name	Value	Units	Tag	Remark
1	Record sequence number	2	-	rec_seq_num	-
2	1st record sub-type code	219	-	rec_subtype_code_1	-
3	Record type code	192	-	rec_type_code	-
4	2nd record sub-type code	18	-	rec_subtype_code_2	-
5	3rd record sub-type code	18	-	rec_subtype_code_3	-
6	Length of this record	360	-	rec_len	-
7	ASCII/EBCDIC Flag	A	-	ascii_ebcdic_flag	-
8	Blanks	-	-	blanks	-
9	Referenced file number	1	-	refl_num	-
10	Referenced file name	ERS2.SAR.PRILEAD	-	refl_name	-
11	Referenced file class	SARLEADER FILE	-	refl_class	-
12	Referenced file class code	SARL	-	refl_class_code	-
13	Referenced file data type	MIXED BINARY AND ASCII	-	refl_dtype	-
14	Referenced file data type code	MBAA	-	refl_dtype_code	-
15	Number of records in referenced file	6	-	nb_rec_in_refl	-
16	Referenced file in - 1st record length	720	-	refl_first_rec_len	-
17	Referenced file maximum record length	12288	-	refl_max_rec_len	-
18	Referenced file record length type	VARIABLE LEN	-	refl_rec_len_type	-
19	Referenced file record length type code	VARE	-	refl_rec_len_type_code	-
20	Referenced file physical volume start nu mber	1	-	refl_phys_vol_start_num	-
21	Referenced file physical volume end numb er	1	-	refl_phys_vol_end_num	-
22	Referenced file portion start, 1st recor d number for this physical volume	1	-	refl_portion_start_first	-
23	Referenced file portion end, last record number for this physical volume	6	-	refl_portion_end_last	-
24	File pointer spare segment	-	-	fl_ptr_spare_segment	-
25	Local use segment	-	-	local_use_segment	-

Record name................: Data Set File Pointer Record

Pos	Esa field name	Value	Units	Tag	Remark
1	Record number	3	-	rec_seq_num	-
2	1-st record subtype code	219	-	rec_subtype_code_1	-
3	Record type code	192	-	rec_type_code	-
4	2-nd subtype code	18	-	rec_subtype_code_2	-
5	3-rd subtype code	18	-	rec_subtype_code_3	-
6	Length of this record	360	-	rec_len	-
7	ASCII/EBCDIC Flag for referenced file	A	-	ascii_ebcdic_flag	-
8	Blank	-	-	blanks	-
9	Referenced file number	2	-	refl_num	-
10	Referenced file name	ERS2.SAR.PRIIMGY	-	refl_name	-
11	Referenced file class	IMAGERY OPTIONS FILE	-	refl_class	-
12	Referenced file class code	IMOP	-	refl_class_code	-
13	Referenced file data type	MIXED BINARY AND ASCII	-	refl_dtype	-
14	Referenced file data type code	MBAA	-	refl_dtype_code	-
15	Number of records in referenced file	8202	-	nb_recs_in_refl	-
16	Referenced file 1-st record length	16012	-	refl_first_rec_len	-
17	Referenced file maximum record length	16012	-	refl_max_rec_len	-
18	Referenced file record length type	FIXED LENGTH	-	refl_rec_len_type	-
19	Referenced file record length type code	FIXD	-	refl_rec_len_type_code	-
20	Referenced file physical volume start nu mber	1	-	refl_phys_vol_start_num	-
21	Referenced file physical volume end numb er	1	-	refl_phys_vol_end_num	-
22	Referenced file portion start, 1-st reco rd number for this physical volume	1	-	refl_portion_start_first	-

Pos	Esa field name	Value	Units	Tag	Remark
23	Referenced file portion end, last record number for this physical volume	8202	-	refl_portion_end_last	-
24	File pointer spare segment	-	-	fl_ptr_spare_segment	-
25	Local use segment	PRI	-	local_use_segment	-

Record name................: TextRecord

Pos	Esa field name	Value	Units	Tag	Remark
1	Record sequence number	4	-	rec_seq_num	-
2	1-st record subtype code	18	-	rec_subtype_code_1	-
3	Record type code	63	-	rec_type_code	-
4	2-nd subtype code	18	-	rec_subtype_code_2	-
5	3-nd subtype code	18	-	rec_subtype_code_3	-
6	Length of this record	360	-	rec_len	-
7	ASCII/EBCDIC Flag	A	-	ascii_ebcdic_flag	-
8	Continuation flag (*)	-	-	continuation_flag	-
9	Product type specifier	PRODUCT:ERS-2.SAR.PRI	-	prod_type_specifier	-

Appendix 2: Media Analysis Example Output

Note also that the product recognizing operation is based on the correlation of the file structure of the media with a set of predefined values. In case of products having a very similar structure, it is obviously not possible for the media content analysis to distinguish them and a possibly wrong recognition is then obtained. The following test is an example of the output of the media analysis output. As can be seen, the media analysis file is divided into two sections. In the first, a lists of the recognized products is present with the three best choices (the most likely comes as first). In the second section, the media structure is shown, in which the number of volumes, of files, number of records an size in bytes is detailed.

```
--------------------------------------------------------
Number of Volume(s) = 1
Product Type       = PRI
Sensor Id          = ERS2
Data Format        = CEOS
Source Id          = DEP
--------------------------------------------------------
Number of Volume(s) = 1
Product Type       = PRI
Sensor Id          = ERS1
Data Format        = CEOS
Source Id          = ESP
--------------------------------------------------------
Number of Volume(s) = 1
Product Type       = PRI
Sensor Id          = ERS2
Data Format        = CEOS
Source Id          = ESP
--------------------------------------------------------
Number of Volume(s) = 1
--------------------------------------------------------
VOLUME: 1

FILE: 1   Number of Record(s)    Record Size
RECORD:                  4          360

FILE: 2   Number of Record(s)    Record Size
RECORD:                  1          720
RECORD:                  1         1886
RECORD:                  1         1620
RECORD:                  1         1046
RECORD:                  2        12288

FILE: 3   Number of Record(s)    Record Size
RECORD:               8202        16012

FILE: 4   Number of Record(s)    Record Size
RECORD:                  1          360
--------------------------------------------------------
```

Appendix 3: Ancillary Data Dump Output and Annotations

A sample of a file obtained as output from this function is shown here.

```
=================================================================
=========================
STB - Sar Tool Box - Telespazio / ESA - ANNOTATION DUMP
=================================================================
=========================
Image: cfvr.s1
=================================================================
=========================
    0image_width45
    1image_length25
    2bits_per_sample32
    3compression1
    4photometric_interpretation1
    5sample_per_pixel1
    6x_print_resolution300.000000
    7y_print_resolution300.000000
    8resolution_unit2
    9tile_width128
   10tile_length128
   11tile_offset24
   12tile_byte_count65536
   13sample_format4
   14dispositionx
   15absolute_calib_k999978.000000
   16antenna_boresight20.355000
   17antenna_elevation_gain_flag0
   18bottom_left_lat51.914104
   19bottom_left_lon6.062989
   20bottom_right_lat51.925541
   21bottom_right_lon5.985325
   22centre_geodetic_lat52.349888
   23centre_geodetic_lon6.195046
   24early_zero_fill_record_number0
   25late_zero_fill_record_number0
   26cross_dopp_freq_const150.656296
   27cross_dopp_freq_linear61187.777344
   28cross_dopp_freq_quad0.000000
   29day_data_point4
   30ellipsoid_semimajor_axis6378.144043
   31ellipsoid_semiminor_axis6356.758789
   32incid_angle_centre_range23.069057
   33line_spacing150.000000
   34map_proj_descrGround range
   35month_data_point8
   36nom_nb_looks_azim3.000000
   37normalisation_ref_range847.000000
   38pixel_spacing125.000000
   39radar_wavelen0.056565
   40replica_power154641.000000
   41sampling_rate18.959999
   42scene_ref_numORBIT: 1508 - FRAME: 2547
   43second_of_day37980.000000
   44time_interval_data_point60.000000
   45top_left_lat52.773964
   46top_left_lon6.408101
   47top_right_lat52.785542
   48top_right_lon6.328794
   49year_data_point1995
   50zero_dopp_azim_first_time04-AUG-1995 10:35:02.322
   51zero_dopp_azim_last_time04-AUG-1995 10:35:17.687
   52zero_dopp_range_first_time5.564405
   53spread_loss_comp_flag1
   54nb_data_points5
   55log_vol_idERS2.SAR.PRI
   56gr_sr_pol_degree4
   57gr_sr_coeff_10.000969
   58gr_sr_coeff_2526.446899
   59gr_sr_coeff_311.997012
   60gr_sr_coeff_4-0.061625
   61gr_sr_coeff_5-0.000199
   62near_zero_fill_pixel_number0
   63far_zero_fill_pixel_number68
   64prf1679.902344
   65x_sat_13569723.600000
   66x_sat_23954408.780000
   67x_sat_34323215.080000
   68x_sat_44674652.340000
   69x_sat_55007300.920000
   70y_sat_1880854.570000
   71y_sat_2824210.230000
   72y_sat_3761068.490000
   73y_sat_4691819.530000
   74y_sat_5616890.060000
   75z_sat_16138981.080000
   76z_sat_25907244.760000
   77z_sat_35652398.400000
   78z_sat_45375435.140000
   79z_sat_55077435.060000
   80vx_sat_16535.191690
   81vx_sat_26283.327860
   82vx_sat_36006.074500
   83vx_sat_45704.555740
   84vx_sat_55379.997550
   85vy_sat_1-887.918640
   86vy_sat_2-999.248640
   87vy_sat_3-1104.391740
   88vy_sat_4-1202.721190
   89vy_sat_5-1293.646730
   90vz_sat_1-3664.545220
   91vz_sat_2-4057.490410
   92vz_sat_3-4434.628530
   93vz_sat_4-4794.478790
   94vz_sat_5-5135.625370
   95subimg_top_left_row3
   96subimg_top_left_col2
   97proc_history
 LOCAL STATISTIC 17-Jan-1998 11:15:13.000  Local Cfvr
   98pixel_typeAmplitude
   99calib_const_appli_flag0
  100adc_satur_compens_flag0
  101chirp_average_density0.000000
  102processing_pafIP
  103processor_nameSAR ERS
  104scaling_factor1.000000
  105x_scale_factor0.100000
  106y_scale_factor0.083333
  107prf_equivalent19.524895
  108doppl_centr_cub_coeff0.000000
  109replica_power_comp_flag0
  110image_scaleLINEAR
  111data_formatceos
  112source_idukp
  113number_of_volumes3
  114row_transient0
```

115col_transient0
==

The following table explains the set of the annotations maintained by the various STB tools.

Table 1:

Annotation name	Meaning	Example value
absolute_calib_k	calibration constant value (linear)	999978.000000
adc_satur_compens_flag	flag indicating if the ADC saturation compensation has been applied (1 means applied)	0
antenna_boresight	boresignt angle (degrees)	20.355000
antenna_elevation_gain_flag	flag indicating if the antenna pattern correction is apllied (1 means applied)	0
bits_per_sample	the number of bits of the pixel of each layer of the image	32
bottom_left_lat	latitude of the last line first pixel corner (degree)	51.914104
bottom_left_lon	longitude of the last line first pixel corner (degree)	6.062989
bottom_right_lat	latitude of the last line last pixel corner (degree)	51.925541
bottom_right_lon	longitude of the last line last pixel corner (degree)	5.985325
calib_const_appli_flag	flag indicating if the spread-ing calibtration constant has been applied (1 means applied)	0
centre_geodetic_lat	latitude of the center (degree)	52.349888
centre_geodetic_lon	latitude of the center (degree)	6.195046
chirp_average_density	density of the chirp replica	0.000000
col_transient	internal TTIF flag	0
compression	internal TTIF flag	1

Table 1:

Annotation name	Meaning	Example value
cross_dopp_freq_const	doppler frequency polynomial order 0 coefficient (Hz)	150.656296
cross_dopp_freq_linear	doppler frequency polynomial order 1 coefficient (Hz/s)	61187.777344
cross_dopp_freq_quad	doppler frequency polynomial order 2 coefficient (Hz/s/s)	0.000000
data_format	format of the SAR product (CEOS or MPHSPH)	CEOS
day_data_point	day in the year of the first state vector	4
disposition	internal TTIF flag	x
early_zero_fill_record_number	number of fill lines at image start	0
ellipsoid_semimajor_axis	ellipsoid semimajor axis (km)	6378.144043
ellipsoid_semiminor_axis	ellipsoid semiminor axis (km)	6356.758789
far_zero_fill_pixel_number	number of filled pixels at end of each image line	68
gr_sr_coeff_1	slant to ground polynomial coeffient 1	0.000969
gr_sr_coeff_2	slant to ground polynomial coeffient 2	526.446899
gr_sr_coeff_3	slant to ground polynomial coeffient 3	11.997012
gr_sr_coeff_4	slant to ground polynomial coeffient 4	-0.061625
gr_sr_coeff_5	slant to ground polynomial coeffient 5	-0.000199
gr_sr_pol_degree	slant to ground polynomial degree	4
image_length	the number of lines of the image	25
image_scale	indication if the image is in LINEAR or DB scale	LINEAR

Table 1:

Annotation name	Meaning	Example value
image_width	the number of pixels of the image	45
incid_angle_centre_range	incidence angle at mid range (degrre)	23.069057
late_zero_fill_record_number	number of fill lines at image end	0
line_spacing	spacing between lines (m)	150.000000
log_vol_id	product identifier string	ERS2.SAR.PRI
map_proj_descr	descriptor of the geographic projection	Ground range
month_data_point	month in the year of the first state vector	8
nb_data_points	number of the state vectors	5
near_zero_fill_pixel_number	number of filled pixels at start of each image line	0
nom_nb_looks_azim	number of looks	3.000000
normalisation_ref_range	reference slant range used for the spreading loss compensa-tion (km)	847.000000
number_of_volumes	number of media volumes	1
photometric_interpretation	internal TTIF flag	1
pixel_spacing	spacing between pixels (m)	125.000000
pixel_type	identificator of the Amplitude or Power or Complex image	Amplitude
prf	Pulse Repetition Frequency (Hz)	1679.902344
prf_equivalent	internal TTIF flag	19.524895
proc_history	processing history sequence	
processing_paf	identification of the process-ing station/PAF	IP
processor_name	identification of the SAR pro-cessing system	SAR ERS
radar_wavelen	wavelenght of the radar signal (m)	0.056565

Table 1:

Annotation name	Meaning	Example value
replica_power	power of the replica chirp	154641.000000
resolution_unit	internal TTIF flag	2
row_transient	internal TTIF flag	0
sample_format	format of the pixel: 1,2 means integer, 4 means floating point representation	4
sample_per_pixel	number of image layers	1
sampling_rate	sampling frequency in range (MHz)	18.959999
scaling_factor	internal TTIF flag	1.000000
scene_ref_num	scene identification string	ORBIT: 1508 - FRAME: 2547
second_of_day	second in the day of the first state vector (s)	37980.000000
source_id	generating station/PAF	IP
spread_loss_comp_flag	flag indicating if the spread-ing loss compensation has been applied (1 means applied)	1
subimg_top_left_col	first line first pixel corner in the entire image coordinate system (column value)	2
subimg_top_left_row	first line first pixel corner in the entire image coordinate system (line value)	3
tile_byte_count	internal TTIF flag	65536
tile_length	internal TTIF flag	128
tile_offset	internal TTIF flag	24
tile_width	internal TTIF flag	128
time_interval_data_point	number of seconds between contiguous state vectors (s)	60.000000
top_left_lat	latitude of the first line first pixel corner (degree)	52.773964
top_left_lon	longitude of the first line first pixel corner (degree)	6.408101

Table 1:

Annotation name	Meaning	Example value
top_right_lat	latitude of the first line last pixel corner (degree)	52.785542
top_right_lon	longitude of the first line last pixel corner (degree)	6.328794
vx_sat_1	state vector velocity 1 (x component)	6535.191690
vx_sat_2	state vector velocity 2 (x component)	6283.327860
vx_sat_3	state vector velocity 3 (x component)	6006.074500
vx_sat_4	state vector velocity 4 (x component)	5704.555740
vx_sat_5	state vector velocity 5 (x component)	5379.997550
vy_sat_1	state vector velocity 1 (y component)	-887.918640
vy_sat_2	state vector velocity 2 (y component)	-999.248640
vy_sat_3	state vector velocity 3 (y component)	-1104.391740
vy_sat_4	state vector velocity 4 (y component)	-1202.721190
vy_sat_5	state vector velocity 5 (y component)	-1293.646730
vz_sat_1	state vector velocity 1 (z component)	-3664.545220
vz_sat_2	state vector velocity 2 (z component)	-4057.490410
vz_sat_3	state vector velocity 3 (z component)	-4434.628530
vz_sat_4	state vector velocity 4 (z component)	-4794.478790
vz_sat_5	state vector velocity 5 (z component)	-5135.625370
x_print_resolution	internal TTIF flag	300.000000
x_sat_1	state vector 1 (x component)	3569723.600000

Table 1:

Annotation name	Meaning	Example value
x_sat_2	state vector 2 (x component)	3954408.780000
x_sat_3	state vector 3 (x component)	4323215.080000
x_sat_4	state vector 4 (x component)	4674652.340000
x_sat_5	state vector 5 (x component)	5007300.920000
x_scale_factor	internal TTIF flag	0.100000
y_print_resolution	internal TTIF flag	300.000000
y_sat_1	state vector 1 (y component)	880854.570000
y_sat_2	state vector 2 (y component)	824210.230000
y_sat_3	state vector 3 (y component)	761068.490000
y_sat_4	state vector 4 (y component)	691819.530000
y_sat_5	state vector 5 (y component)	616890.060000
y_scale_factor	internal TTIF flag	0.083333
year_data_point	year of the first state vector	1995
z_sat_1	state vector 1 (z component)	6138981.080000
z_sat_2	state vector 2 (z component)	5907244.760000
z_sat_3	state vector 3 (z component)	5652398.400000
z_sat_4	state vector 4 (z component)	5375435.140000
z_sat_5	state vector 5 (z component)	5077435.060000
zero_dopp_azim_first_time	time of the first image line	04-AUG-1995 10:35:02.322
zero_dopp_azim_last_time	time of the last image line	04-AUG-1995 10:35:17.687
zero_dopp_range_first_time	time of the first image pixel (ms)	5.564405

Appendix 4: AOI Specification

An AOI specified using coordinates, can have the following two shapes:

- a rectangular region
- a polygonal region

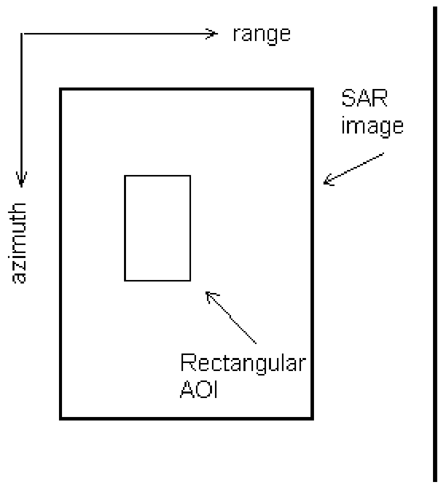


Fig. 17 - A Rectangular AOI

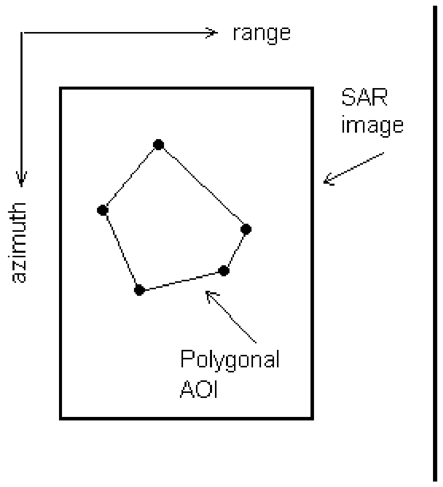


Fig. 18 - A Polygonal AOI

The AOI can be placed in two ways respect the SAR image:

- internal
- partly external

The previous "Fig. 17 - A Rectangular AOI" is related to a internal AOI while the following "Fig. 19 - An AOI partly external to the image" shows an AOI placed partly externally to the image.

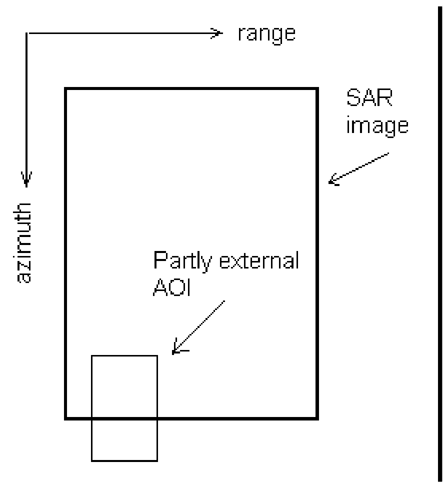


Fig. 19 - An AOI partly external to the image

The Rectangular AOI can be specified in the following forms:

- Top Left Corner and Bottom Right Corner
- Top Right Corner and Bottom Left Corner
- Center and Size

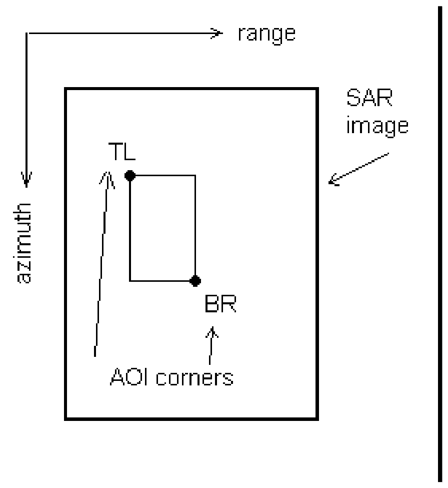


Fig. 20 - Rectangular AOI specified through Top Left, Bottom Right Corners

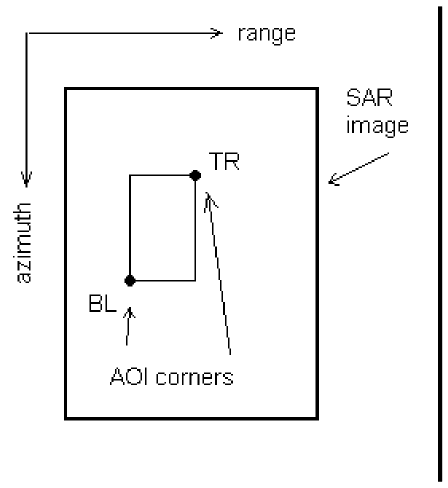


Fig. 21 - Rectangular AOI specified through Top Right, Bottom Left Corners

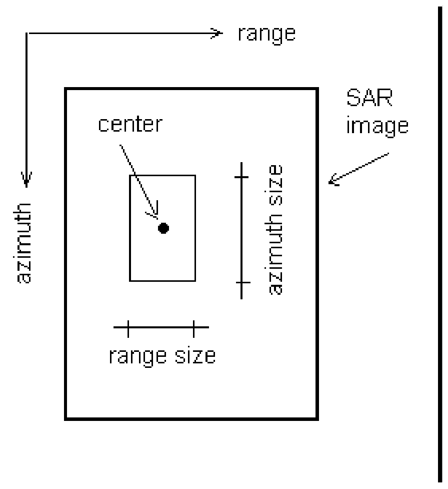


Fig. 22 - Rectangular AOI specified through Center, Size

Both the Top Left, Bottom Right Corners couple and the Top Right, Bottom Left Corners can be specified in one of the two coordinate systems:

- geodetic latitude, longitude
- row,column

The Center point can be specified in one of the two coordinate systems:

- geodetic latitude, longitude
- row,column

The Size can be specified in

- kilometers unit
- pixel units

The polygonal AOI can be specified through the following data:

- the number of the vertices of the polygon
- the coordinates of the vertices

The vertices can be expressed in the following coordinate systems:

- geodetic latitude, longitude
- row,column

The following table summarizes the various specifications of the AOI with the related portions of the ".INI" file.

AOI Specification	Example	Comment

• Rectangular
• Top Left Bottom Right corners
• Row, Column coordinates
	`Coordinate System="ROWCOL"`
	`Top Left Corner=100,200`
	`Bottom Right Corner=300,500`
		• the first coordinate is for row, the second for column
		• the row,col coordinate system is assumed by default if the Coordinate System statement is not present

• Rectangular
• Top Left, Bottom Right corners
• Lat,Lon coordinates
	`Coordinate System="LATLON"`
	`Top Left Corner=52.70,6.30`
	`Bottom Right Corner=52.75,6.35`
		• coordinates expressed in degree units
		• the first coordinate is for lat, the second for lon
		• the coordinate system statement must be present

• Rectangular
• Top Right Bottom Left corners
• Row, Column coordinates
	`Coordinate System="ROWCOL"`
	`Top Right Corner=300,200`
	`Bottom Left Corner=100,500`
		• the first coordinate is for row, the second for column
		• the row,col coordinate system is assumed by default if the Coordinate System statement is not present

• Rectangular
• Top Right, Bottom Left corners
• Lat,Lon coordinates
	`Coordinate System="LATLON"`
	`Top Left Corner=52.78,6.30`
	`Bottom Right Corner=52.75,6.35`
		• coordinates expressed in degree units
		• the first coordinate is for lat, the second for lon
		• the coordinate system statement must be present

• Rectangular
• Center in row,col
• Size in pixel
	`Centre=200,300`
	`SizeUnit="ROWCOL"`
	`Size=100,150`
		• the first coordinate is for row, the second for col
		• the first size is for row, the second for col

• Rectangular
• Center in row,col
• Size in km
	`Centre=50,50`
	`Size Unit="KM"`
	`Size=1.5,2.0`
		• the first size is for km in row direction , the second size is for km in col direction

• Rectangular
• Center in lat,lon
• Size in pixel
	`Coordinate System="LATLON"`
	`Centre=52.460,5.519`
	`Size Unit="ROWCOL"`
	`Size=100,150`	

• Rectangular
• Center in lat,lon
• Size in km
	`Coordinate System="LATLON"`
	`Centre=52.460,5.519`
	`Size Unit="KM"`
	`Size=1.5,2.0`	

• Polygonal
• Vertices in row,col
	`Coordinate System="ROWCOL"`
	`Number of Vertex=4`
	`Ver-`
	`tex=100,250,250,100,400,250,250,`
	`400`
		• the vertices are in the form V_1row,V_1col,V_2row,V_2col,...,$V_n$row,$V_n$col
		• the row,col coordinate system is assumed by default if the Coordinate System statement is not present

● Polygonal
● Vertices in lat,lon
	`Coordinate System="LATLON"`
	`Number of Vertex=4`
	`Vertex=52.78,6.41,52.79,6.34, 52.76,6.32, 52.75,6.39`
		● the vertices are in the form V_1lat,V_1lon,V_2lat,V_2lon,...,$V_n$lat,$V_n$lon
		● the coordinate system statement must be present

Some care must be taken when using an AOI specified through the Top Right and Bottom Left corners in Latitude,Longitude system, for products which are not geocoded (like the RAW, SLC or PRI). Lets look at a SAR image seen it in a geographical projection (see “Fig. 23 - a PRI seen in a geographical chart”). When you select the two corners TL and BR, you are thinking to the AOI shown in “” but due to the fact that the range axis is not parallel neither to the latitude neither to the longitude axes you could obtain the AOI shown in “”. This happens because the system convert the two corners coordinates from lat,lon to row,col and then uses these to extract the related AOI.

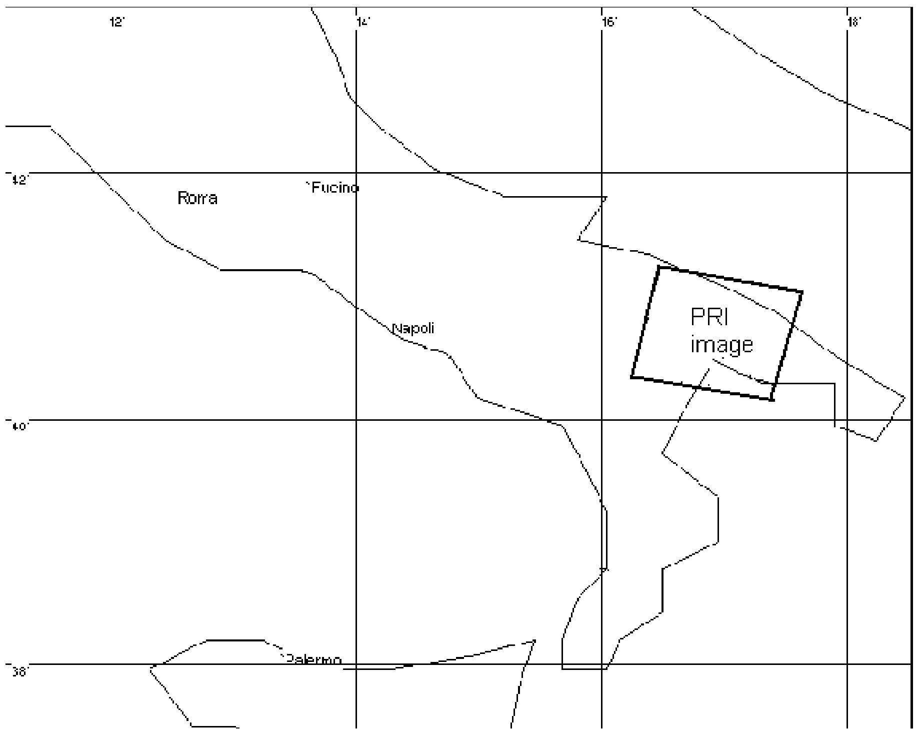


Fig. 23 - a PRI seen in a geographical chart

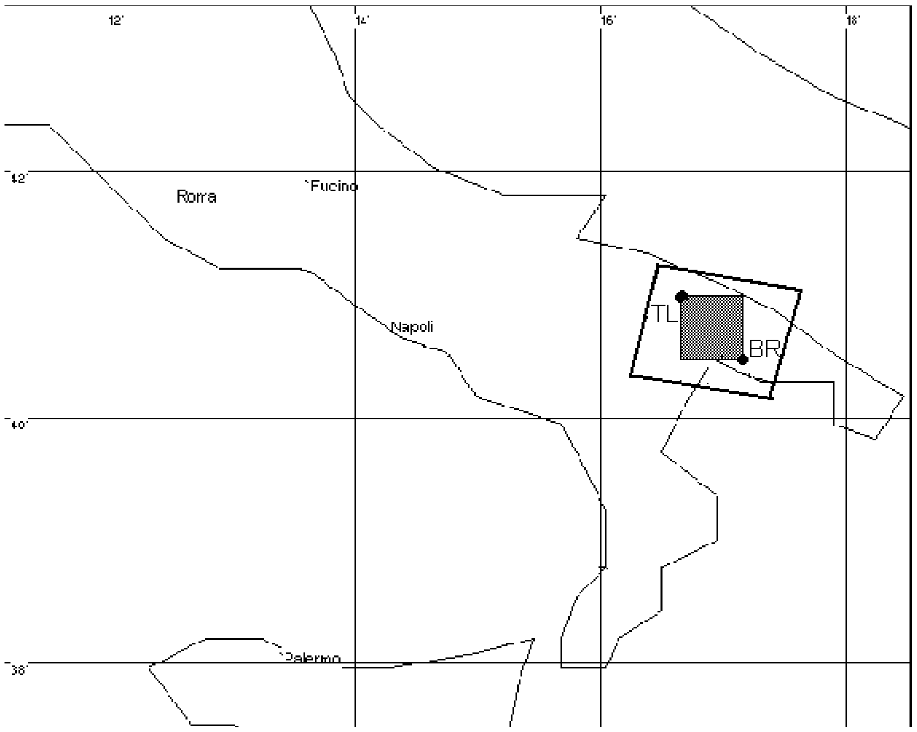


Fig. 24 - What the user was thinking

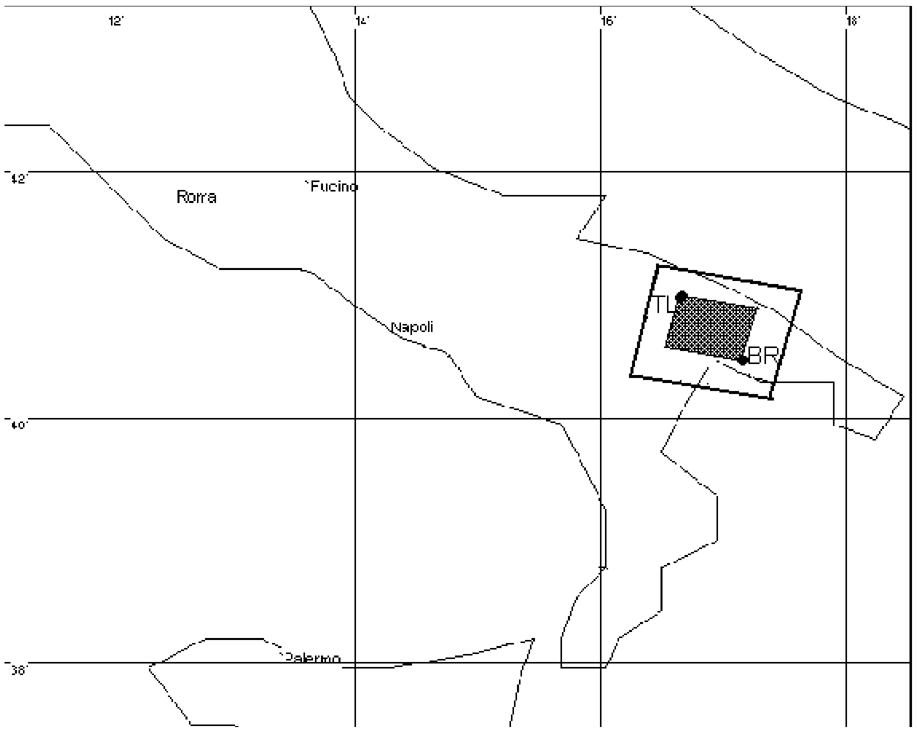


Fig. 25 - What the user could obtain

To avoid these problems an “ad hoc” management of such situations has been developed. For all tools except the Statistical tool with the local statistic, the AOI obtained is the same as in “Fig. 25 - What the user could obtain” (i.e. the rectangular area with sides parallel to the range and azimuth axes having the specified TL and BR corners) and a warning message is issued (see Sect.: “The HMI section of the SAR Toolbox system is devoted to the simple creation of the “.INI” files containing the processing parameters. The 7 tools are in a one to one correspondence with the menus of the main menu bar showed in next figures:”). Due to the capability of the statistical tool to manage polygonal regions, the situation shown in “Fig. 24 - What the user was thinking” does apply (notice that only a polygonal region can have sides not parallel to the range and azimuth axes).

AOI specification by example image

The remaining way to specify the AOI is associated to the possibility to give to the system not the AOI vertices in some coordinate system, but an example image. In other words, an AOI can be specified through an image portion, selected in some way. This approach is limited in the sense that is implemented only in the extraction tool and only for the quicklook image.

Starting from the quicklook image, you can select a portion (using IDL or xv or another image processing system capable to crop an image) and tell to the system to retrieve the related coordinates. In this way you will obtain the Top Left, Bottom Right coordinates in row,col system, expressed in the full resolution reference, ready to be used in the remaining tools, without knowing anything about the coordinates, just seeing the image and visually selecting the AOI.

Appendix 5: Sequential Execution of the Toolbox Algorithms

The SAR Toolbox tools can handle “.INI” files with multiple header sections in order to execute the various functions in a sequential way. Useing this technique the creation of processing chains became a very simple task, requiring just the appending of the various elementary “.INI” files in one unique file and their subsequent activation. Let us suppose that we need to generate from a SLC image portion (already extracted) the related square modulus. This can be obtained with the data conversion tool first converting the complex input to modulus and then raising it to the square.
These two operations can be combined in the following unique “.INI” file (save it with name powmodulus.ini):

```
[COMPLEX TO AMPLITUDE]
Input Dir = "./"
Output Dir = "./"
Input Image = "slcimage.e5"
Output Image = "modulus"
[AMPLITUDE TO POWER]
Input Dir = "./"
Output Dir = "./"
Input Image = "modulus.v2"
Output Image = "power_modulus"
```

executed with the following command:

```
stbx -if powmodulus.ini
```

The stbx tools have another important feature. When the first section in the composite “.INI” file is [GLOBAL SETTING], the related parameters have a global meaning and can be used by all the functions invoked in the “.INI” file.
This section can contain assignments for the following parameters only:

- input directory
- output directory
- temporary directory
- delete input file flag

The previous example can then be transformed in:

```
[GLOBAL SETTING]
Input Dir = "./"
Output Dir = "./"
Temp Dir = "./"
Delete Input Image = 'Y'
[COMPLEX TO AMPLITUDE]
Input Image = "slcimage.e5"
Output Image = "modulus"
[AMPLITUDE TO POWER]
Input Image = "modulus.v2"
Output Image = "power_modulus"
```

If one of the global parameters is also inserted in one of the “.INI” sections, it overrides the global setting but just for such section.
In the following example:

```
[GLOBAL SETTING]
Input Dir = "./"
Output Dir = "./"
Temp Dir = "./"
Delete Input Image = 'Y'
[COMPLEX TO AMPLITUDE]
Temp Dir = "tmp"
Input Image = "slcimage.e5"
Output Image = "modulus"
[AMPLITUDE TO POWER]
Input Image = "modulus.v2"
Output Image = "power_modulus"
```

the temporary directory, globally set to "./", changes to tmp for the complex to amplitude conversion only.

Appendix 6: System Performance and Memory Issues

The parameters which affect the system performance of the SAR Toolbox system, on the various machines are described here. These parameters are all related to the main memory allowable to the SAR Toolbox. This amount, shall be carefully selected for multi user machines like the Sun or the DEC. A too small amount of such memory imply that the SAR Toolbox system dimension its image buffer to a very little extent so requiring many disk access to cover the image processing action involved. On the other hand, if this size exceeds the physical memory allowable for the process the OS began to swap the memory on disk, causing again loss of performances. For single user machines, like PC or Mac use the OS commands to measure the amount of free physical memory. If you are in doubt, select few Mbytes less than the amount of installed physical memory. The system ".INI" file is listed below and is the same for all the machines, so you have to modify just the section related to your computer. The memory amount is expressed in kbytes.

```
[SUN]
System Memory = 16384
[SGI]
System Memory = 16384
[HP]
System Memory = 16384
[OSF]
System Memory = 16384
[IBM]
System Memory = 16384
[DOS]
System Memory = 16384
[WIN95]
System Memory = 16384
[MAC]
System Memory = 16384
```

Appendix 7: IDL Interface Activation

In the following section the activation of the interface between IDL and the SAR Toolbox functions shall be described. Due to the integration of the SAR Toolbox functions in IDL using the "linkimage" command, the SAR Toolbox functions became a part of the IDL system and can be called as any other IDL function. It is supposed that the SAR Toolbox is properly installed and that the symbols to enable the automatic execution files are correctly defined, otherwise no SAR Toolbox will appear present in IDL. You can check this entering in IDL with stbidl and verifying that after the RSI copyright notice, the following message appears:

```
Parameter !ORDER equal to 1
TIFF library is allowable
STBXSHR exported routines:
   stbx_help
```

(follows a list of routine names)

If all works you can have help on the SAR Toolbox function using the following command:

```
stbx_help
```

The following message will appear, explaining how to get more specific help on the selected function:

```
% Usage: stbx_help, "stbx_topic"
% Allowed stbx_topics:
```

(follows a list of routine names)

As an example, try:

```
stbx_help,"stbx_oversampling"
```

You will get the following help:

```
%
Calling Sequence
OutIma = stbx_oversampling( InpIma, TLRow, TLCol, RangeSpacing, FdcPoly, PRF, OutRowSize, OutColumnSize)
Arguments
InpIma
The input image to resample
TLRow
Top left row of the first image pixel in the full image reference frame
TLCol
The first image pixel in the full image reference frame
RangeSpacing
Spacing in the range direction (take it from image annotations)
FdcPoly
Array with the coefficients of the doppler centroid frequency polynomial (take them from image annotations)
PRF
The Pulse Repetition Frequency of the image
OutRowSize
The row size of the output image
OutColumnSize
The column size of the output image
OutIma
Output image always of FLOAT data type or COMPLEX for complex input images
```

Appendix 8: The HMI System

As seen, the SAR Toolbox System functioning is based on the usage of parameter files called “.INI” files. These files are pure ASCII and can be created using an editor like the DOS edit or the UNIX vi. To simplify this operation a Human Machine Interface has been developed, which assists in this phase allowing:

a clear indication of the various parameters needed by each tool, with an indication if the parameter is mandatory or optional
the input parameters value checking against limit values, allowing the generation of valid “.INI” files
the filling of the parameters with default values avoiding the user to remember
the saving and loading of the “.INI” files, allowing the creation of an archive of such files, to be used with small changes in similar processing situations
The procedures for the HMI activation are slight different depending on the machines.
UNIX-The HMI can be activated, with the stbhmi command contained in the sub-directory /bin of the SAR Toolbox root dir. No icon is foreseen for the activation
PC WIN95-The HMI can be activated by double clicking on the file contained in the subdirectory \bin of the SAR Toolbox root dir and named stbhmi.tcl. To create an icon for the activation make a link with this file using the Win95 commands
MAC-The HMI can be activated by double clicking on the icon named Wish then using Source item on File menu opening the stbhmi.tcl file inside cfg folder inside Sar Toolbox folder

When activated the following menu bar shall appear:

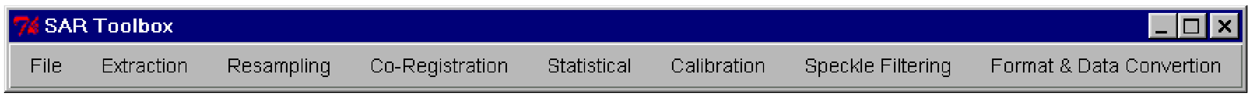


Fig. 12 - The Toolbox HMI menu bar

Each menu (except the File which allow just the exit from the HMI) has a direct reference to one tool. The following example shows how to execute a resampling of a full resolution image portion already extracted, using the HMI.

Activate the Resampling menu. The following menu shall appear:

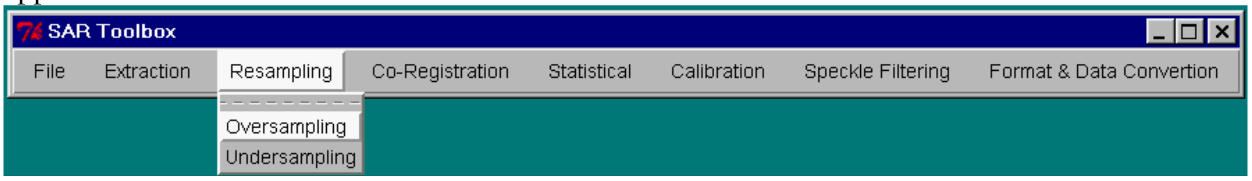


Fig. 13 - The Resampling menu

When you choose Oversampling the following window will appear:

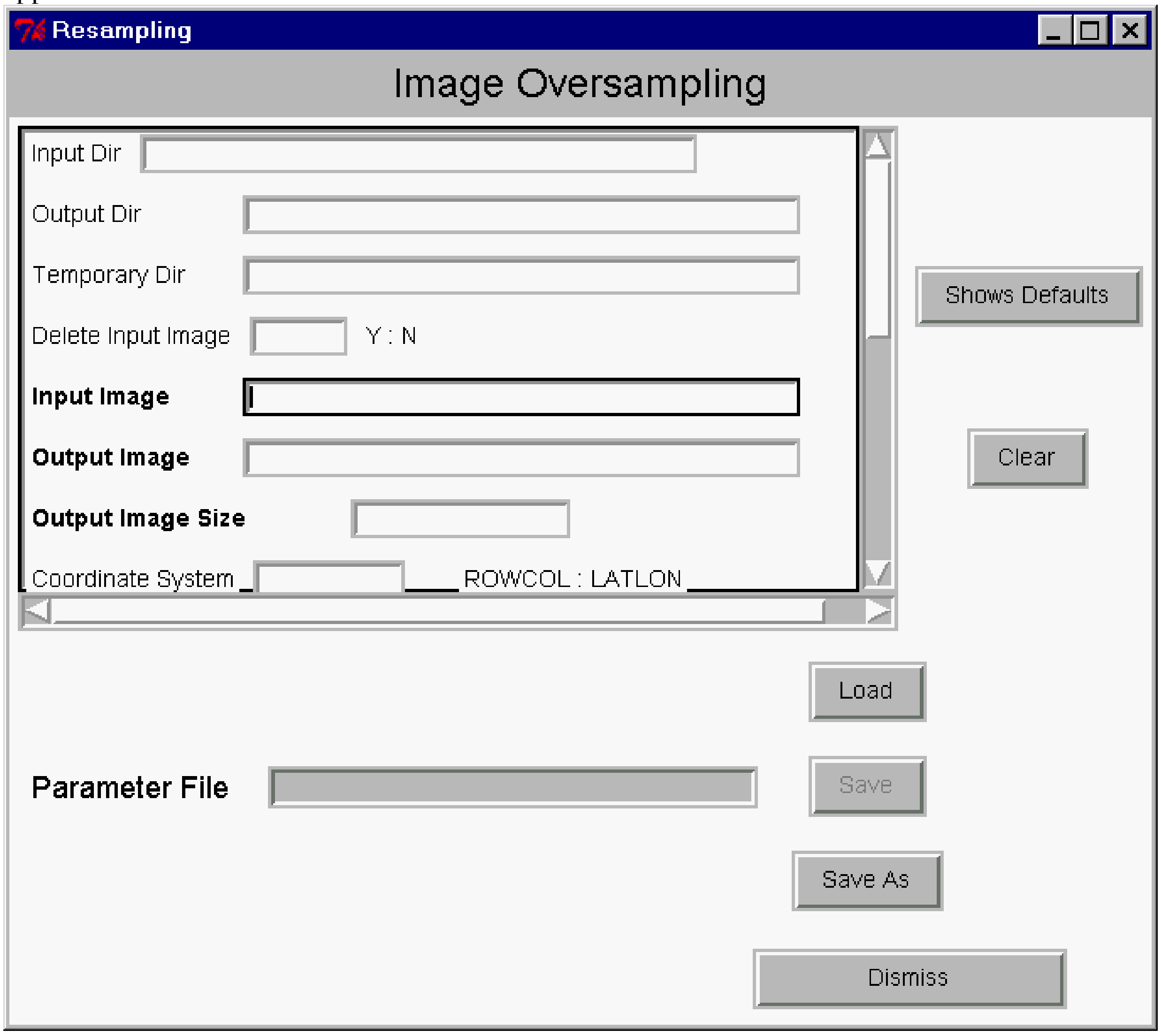


Fig. 14 - The image oversampling window

In this window you can see many a scrollable region containing many fields, six buttons and a parameter field section. In the field region you can distinguish among others, only few fields having a name in bold text style. These fields are the mandatory fields you must specify in order to run the toolbox function.

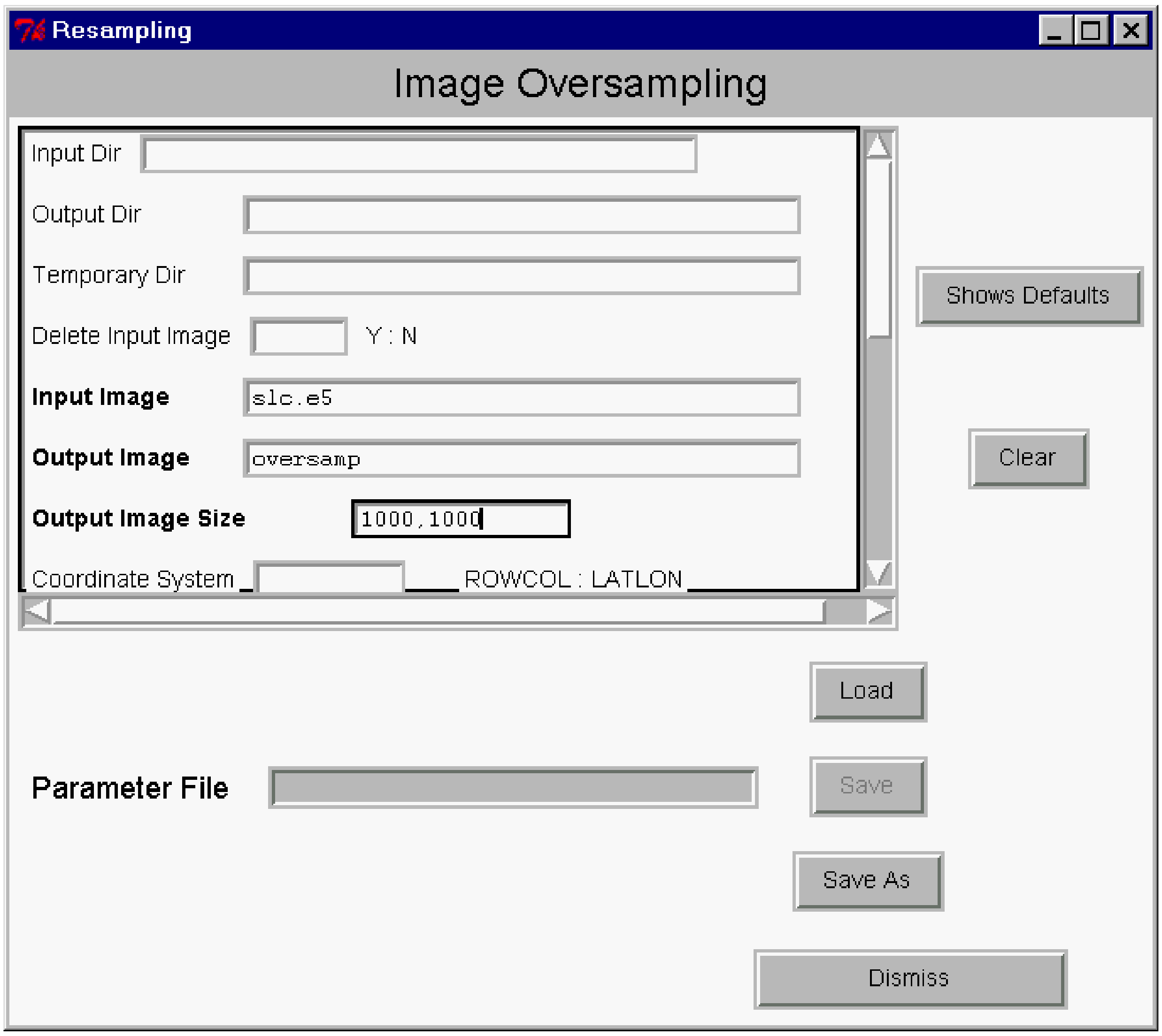
In this case they are:

Input Image, used to specify the name of the image you want to oversample

Output Image, used to specify the name of the image obtained after the oversampling operation

Output Image Size, used to specify the enlarged size of the output image

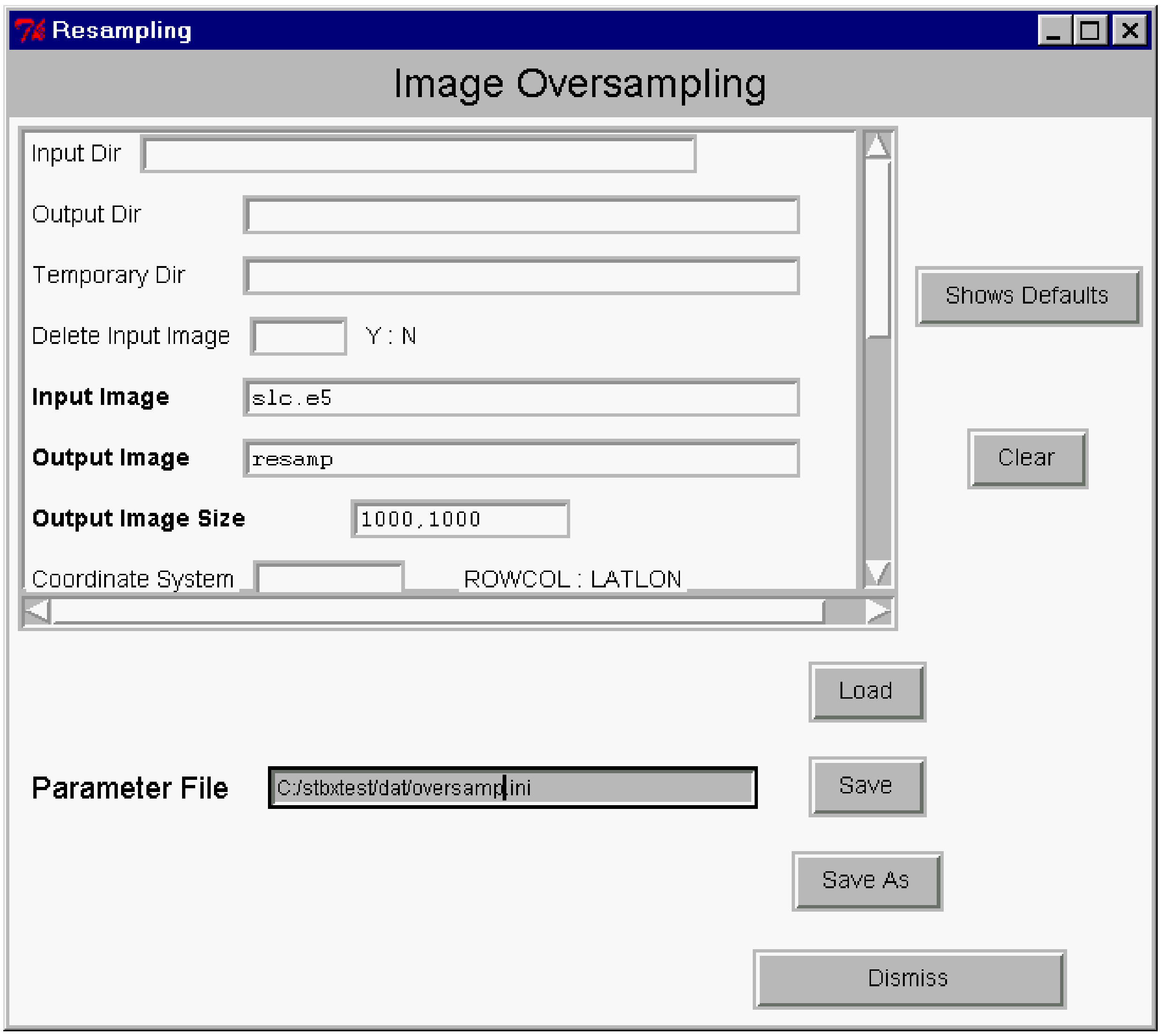
When these parameters are filled with the values the following window shall appear:



Input Dir
Output Dir
Temporary Dir
Delete Input Image Y : N
Input Image slc.e5
Output Image oversamp
Output Image Size 1000,1000
Coordinate System ROWCOL : LATLON
Parameter File

Fig. 15 - The Oversampling window with the selected parameters

Now it is possible to save these parameters into the ".INI" file using the Save As command. This will open a File Selector window allowing you the specification of the ".INI" file name (try as an example oversamp.ini). After this operation the following situation shall appear:



Input Dir
Output Dir
Temporary Dir
Delete Input Image Y : N
Input Image slc.e5
Output Image resamp
Output Image Size 1000,1000
Coordinate System ROWCOL : LATLON
Parameter File C:/stbxtest/dat/oversamp.ini

Fig. 16 - The Oversampling window after a Save As operation

As you can notice the Parameter File field has changed and now contain the name of the saved file. If you change some processing parameter and you want to save the result, just press the Save button.

Another way to perform the saving operation is based on the Parameter File field in conjunction with the Save button. Instead using the Save As button you can specify directly here a file name and then just press the Save button (no file selector will appear). This possibility can be useful when a series of repeated modifications of the parameters shall be done, with a fast saving always with the same name.

The resulting “.INI” file which is created with this operation is the following:

```
[IMAGE OVERSAMPLING]

Input Image = "slc.e5"

Output Image = "resamp"

Output Image Size = 1000,1000
```

The remaining most useful function are associated to the possibility to load a previously generated file (Load button), to clear all the parameters (Clear button) and to exit from this window (Dismiss button).

Appendix 9: The SAR Toolbox Internal Format

The internal format adopted in STB is called TTIFF which stands for Tiled Tagged Image File Format. The TTIFF format is a particular form of the TIFF format and is very similar to the internal format of the Italian PAF products (which is called BTIFF, Blocked Tagged Image File Format). The slight differences are essentially associated with the name of some image parameters (which, in the TIFF world, are called “tags”) and with some restrictions in the image organization. To clarify the topic, a brief description of TIFF, BTIFF and TTIFF formats follows:

The TIFF format is a image file format used mostly for the PC Desktop Publishing applications for encouraging the exchange of digital images between the various packages. The main features of the TIFF format are:

capable to describing bilevel, greyscale, palette-color and full RGB color image data in several color spaces

includes a number of image compression schemes

is not tied to specific hardware

is portable; it does not favor particular operating systems, file systems, compiler or processors

is designed to be expandable and evolve as new needs arise

allows the inclusion of an unlimited amount of private or special-purpose information

allows the presence of any number of images in one file, each with his own set of annotations

A TIFF file is logically divided in two sections: an annotations data part and an image data part. The annotations are stored and retrieved in a TIFF file by their name, i.e. the tag number. This number is used as a entry for a table (called IFD, Image File Directory) of pointers that show the zone of the TIFF file in which the annotations is kept (see Figure 2). In this way a change of the position of the parameters (or the adding of further ones) does not affect the capability to retrieve the data (because all the read operations use this indexed mechanism). For these reasons the TIFF format has a high tolerance to the annotations evolution (position change, new fields, change of datatype for an annotation, and so on).

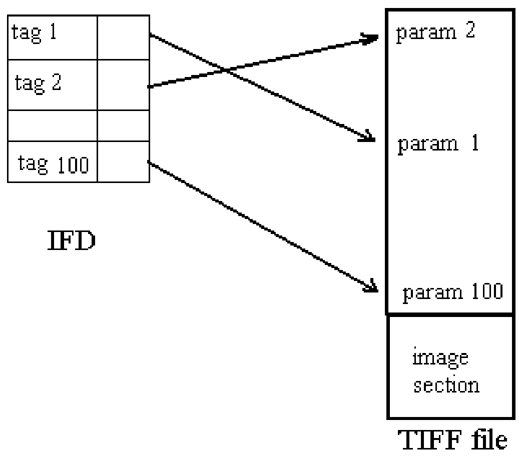


Figure 2

The image section is kept in the TIFF as a sequence of strips (see Figure 3) that have the same columns as the original image and contains a number of rows (this number is chosen to obtain strips of a given size like 8 Kbytes, 16 Kbytes and so on). The location of the strips is stored in the same indexed way as for the image parameters. In this way the image can be easily accessed in subsections and moreover, when the compression is applied, a strip can be efficiently treated by the compression SW.

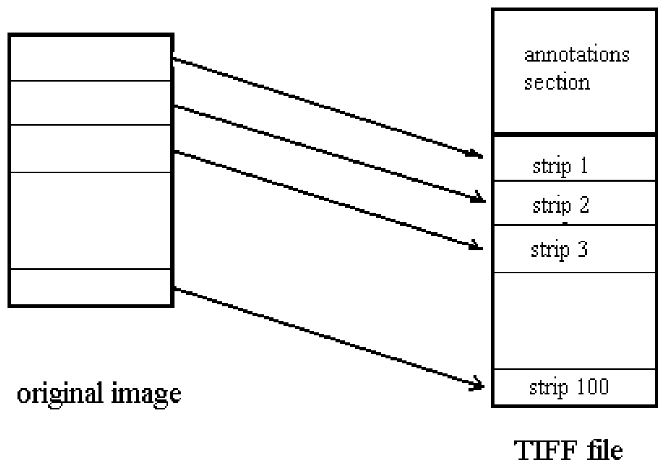


Figure 3

The BTIFF (Blocked TIFF) is an evolution of this format and has the following characteristics: the image is kept on file as tiles (instead of strips) to improve the image access efficiency independently of the access direction and the position of the sub-image accessed (see Figure 4).

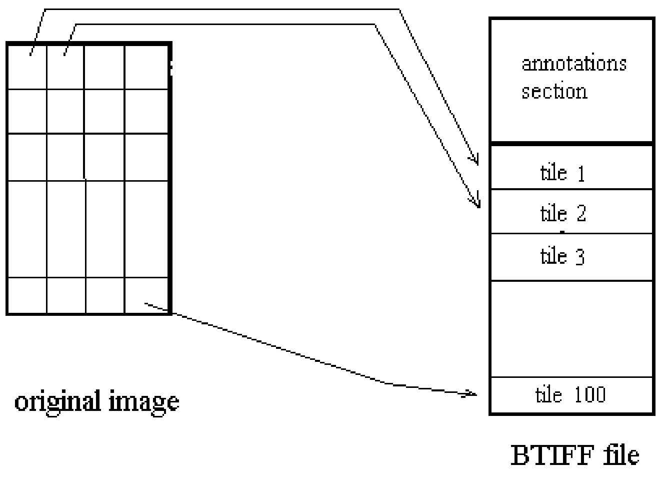


Figure 4

The handling library does all the job and the user does not have to worry about this structure (he can read an image portion in the same way he reads an entire RASTER file). Like TIFF, the image and the parameters are kept in the same file and in a non positional format (i.e. an index table is used). If the format or the number of the parameters is changed there is always a backward compatibility and moreover the handling library does not have to be changed (in this way a "dynamic" format can be handled). Because the BTIFF is a particular subset of the TIFF, the compatibility is maintained and it is possible to import and export images in TIFF with the same routines as the BTIFF one. The BTIFF has no limits in the handling of the multi-band data and/or having a pixel size different from 8 bit (e.g. complex images and so on). It is possible to compress the image via the LZW algorithm. The parameters can be maintained in their natural format e.g. the strings are kept as strings and the numbers as their binary representation (and NOT as strings too).

Efficiency in read-write operations with respect to the memory buffer usage is granted by tuning two parameters which controls the row and column dimensions of the tile, i.e. the elementary unit of the image. At IPAF we use a tile of 100 pixels 100 lines but different values can be used (and this is both transparent to the user and maintains the compatibility).

New TIFF specifications (TTIFF) were issued in parallel to the development of the BTIFF format. These are largely compatible, but differ at a few points:

the name of some parameters are different (e.g. BLOCKOFFSET has the tag 324 in the TTIFF and 273 in the BTIFF

the number of rows and columns of the tile are constrained to be a multiple of 16 for TTIFF (no limitations for the BTIFF)

in the BTIFF the image can be stored with the tiles organized horizontally or vertically (this feature does not exists in TTIFF)

Because the TTIFF is now a standard for many image processing packages (like XV for UNIX or ULEAD for PC) and because the handling library is exactly the same, the TTIFF has been selected as the internal format of STB. This will permit the direct ingestion of the STB images into display SW which accept as input the TTIFF format.

The unique limitation of TTIFF compared to BTIFF is the loss of the possibility to store the tiles vertically (which is of some utility only in the case of a vertical image elaboration, e.g. like the azimuth compression in SAR processing) but is surpassed by the advantage of the direct ingestion of the STB internal images in the image processing SW, without any conversion to standard TIFF.

However, to permit the visualization of the STB images under old SW which does not have the TTIFF ingestion capability, the STB will allow the generation of non tiled standard TIFF images with some little modifications both to the tags concerning the tile dimensions and a reformatting of the image in order to transform tiles into strips.

In case of complex and other non 8-bit images a transform to a single 8bit image is performed.

IDL is capable of reading TIFF images using the *TIFF_READ* command having the following syntax:

Result = TIFF_READ(*File [, R, G, B]*)

where *File* is the file name of the image, *R*, *G* and *B* are optional vectors used to store the look-up table of a Palette color image and *Result* is a two-dimensional matrix containing the image pixels. If the TIFF image is a RGB true color one, *Result* will be a three-dimensional matrix holding in plane 0, 1 and 2 each one of the RGB components.

ERMAPPER includes an import menu to load a TIFF image and transform it into its internal format (see next paragraph). This option can be also activated via the operating system shell with the following command:

importmany *TIFF-Image-File ERMAPPER-Image-File*

The TIFF gray-level image files are transformed into a one band ERMAPPER file, while both RGB true color and Palette color images are always transformed into three band ERMAPPER image files.