Guide to ASAR Geocoding

ESRIN Contract No. 20907/07/I-EC

Authors: David Small & Adrian Schubert

Distribution List	
Name	Affiliation
Nuno Miranda	ESA-ESRIN
Betlem Rosich	ESA-ESRIN
Emmanuel Cérou	ESA-ESRIN
David Small	RSL
Adrian Schubert	RSL
Erich Meier	RSL

Document Change Record			
Issue	Date	Page(s)	Description of the Change
0.91	09.02.2008	all	Initial Draft Issue
0.98	28.02.2008	all	Revisions after initial ESA review of draft
0.981	03.03.2008	most	Fixed fuzzy equations
1.0	19.03.2008	most	Language flow adjustments Further details for GM1 & orbit interpolations Figure 4 as vector graphic Revisions after final ESA comments on draft Removal of draft status
1.01	30.04.2008	13, 14, 16, 20, 30	Formatting; add GM1 to Figure 1; column width in Table 8; phrasing; add CFI library reference

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The ability to geolocate ENVISAT ASAR image products and transform them into a map projection is a critical step required to enable not only overlays with other sources of information (e.g. DEM, GIS layers), but even with other ASAR products, especially those acquired with a different track, beam, or incidence angle.

This document describes methodologies to geocode ASAR images that present themselves in a single 2D raster radar geometry (slant or ground range). It has been written for ESA to provide a reference for developers that wish to develop software to geocode SAR products stored in the ENVISAT format. At the time of this writing, ERS-1, ERS-2, and ASAR products are available in the "ENVISAT" format.

Within this document, two geocoding algorithms are differentiated: the first uses available ENVISAT state vector information together with the radar timing annotations (as well as slant/ground range transformation parameters in the case of ground range products) to geolocate – this method is called Range Doppler (referred to as "RD"). The second simpler method uses instead a rough geolocation grid annotated within the product – this method is referred to simply as "GG".

1.1 ASAR Product Types for Geocoding

The ASAR products that can be geocoded with the methodology described here are listed in Table 1. The three-letter abbreviation for each product type is highlighted in bold face.

Table 1: ASAR Product Types input to geocoding

Acquisition Mode	Product Type			
	Slant Range Complex	Ground Range Detected Precision	Ground Range Detected Medium Resolution	Ground Range Global Monitoring 1km Resolution
IM	ASA_**IMS**_1P	ASA_**IMP**_1P	ASA_**IMM**_1P	-
AP	ASA_**APS**_1P	ASA_**APP**_1P	ASA_**APM**_1P	-
WS	ASA_**WSS**_1P	-	ASA_**WSM**_1P	-
GM	-	-	-	ASA_**GM1**_1P

Two additional product types, ASA_**IMG**_1P and ASA_**APG**_1P, may also be produced by the PF-ASAR processor. They are classified as "Geocoded-Ellipsoid-Corrected" (GEC) products, and are generated using an ellipsoid-geocoding methodology. No elevation model with consideration of local terrain heights is applied during their geolocation. They are presented in a map projection (e.g. UTM), not the slant or ground range geometry common to all the products listed in Table 1.

This document describes how to geocode an ASAR product in the general "Geocoded-Terrain-Corrected" (GTC) sense, whereby a DEM is consulted to make use of knowledge of variations in local height values during geolocation. If no DEM is available, a single mean value may be used during geolocation of each point, and a "GEC" product is produced rather than the more general "GTC" output image. Radiometric normalisation of local terrain effects, i.e. "Radiometrically-Terrain-Corrected" (RTC) products [13], are not considered in this document.

1.2 Acronyms

Table 2: Document Acronyms

ADSR	Annotation Data Set Record in ASAR product annotations
AGP	Antenna Gain Pattern
ALE	Absolute Location Error
AP	Alternating Polarisation
APG	Alternating Polarisation Geocoded ellipsoid corrected
APM	Alternating Polarisation Medium resolution product type
APP	Alternating Polarisation Precision product type
APS	Alternating Polarisation Single look complex product type
ASAR	Advanced Synthetic Aperture Radar on ENVISAT
ASCII	American Standard Code for Information Interchange
CHOM	Swiss Oblique Mercator map projection type
DEM	Digital Elevation Model
DN	Digital Number
DSD	Data Set Descriptor – describes ASAR product annotation structure
DSR	Data Set Record in ASAR product annotations
GADS	Global Annotation Data Set in ASAR product annotations
GEC	Geocoded Ellipsoid Corrected (*single* ellipsoid height for whole scene considered for geolocation)
GG	Geolocation Grid
GTC	Geocoded Terrain Corrected (2D raster of *local* height values considered for geolocation)
IM	Image Mode
IMG	Image Mode Geocoded ellipsoid corrected
IMM	Image Mode Medium resolution product type
IMP	Image Mode Precision product type
IMS	Image Mode Single look complex product type
ITRF	International Terrestrial Reference Frame
LADS	Localisation geolocation grid Annotation Data Set in ASAR product annotations
LCC	Lambert Conformal Conic map projection type
LUT	Look-Up Table
MDS	Measurement Data Set (ASAR product structure)
MJD	Modified Julian Day

MPP	Main Processing Parameters (ASAR product annotation)
PF-ASAR	ASAR Processing Facility
RD	Range Doppler
RSL	Remote Sensing Laboratories - University of Zürich, Switzerland
RTC	Radiometrically Terrain Corrected (2D raster of height values considered for geolocation and *radiometric normalisation* for local terrain variations)
SLC	Single-Look Complex data set
SPH	Specific Product Header (ASAR product annotation)
UPS	Universal Polar Stereographic map projection type
UTM	Universal Transverse Mercator map projection type
WGS84	World Geodetic System 1984
WS	Wide Swath
WSS	Wide Swath Single look complex product type
WSM	Wide Swath Medium resolution product type
ZDT	Zero Doppler Time

1.3 PF-ASAR Data Types

A set of data type primitives used within the PF-ASAR product annotations is listed in Table 3. The data type names listed there are used in later references to the PF-ASAR annotation format.

Table 3: PF-ASAR Data Type Primitives

Data Type	Description
`Ad`	ASCII double: a double-precision floating-point number, expressed as a text string
`Afl`	ASCII float: a single-precision floating-point number, expressed as a text string
`Al`	ASCII long: a 4-byte long integer, expressed as a text string
`fl`	float: a single-precision floating-point number
`mjd`	Modified Julian Day: three consecutive long integers (day, seconds, μseconds) specifying a date and time
`sl`	signed long: a 4-byte long integer
`ul`	unsigned long: a 4-byte unsigned long integer

2 REQUIRED SAR GEOMETRY PARAMETERS

PF-ASAR products are formatted as a set of several ASCII text sections describing the most important global parameters, followed by more specific information in mixed ASCII-binary format, and finally, the binary Data Set Records (DSRs) themselves. The ASCII sections consist of the Main Product Header (MPH), a Specific Product Header (SPH) and a number of Data Set Descriptors (DSDs).

The parameters potentially used during the geocoding of a PF-ASAR product are listed in Table 4 to Table 7. Descriptions of selected data types are listed in Table 3. Product data format field name and table numbers refer to [3], except for references to the DSD format, described in [8]. Subscripts denoting the *sub-swath* (for WSS) or *polarisation* (for AP products) in question are used when necessary. When several values are defined (e.g. 11 floating-point values), a subscripted index denotes the particular value, beginning at 1. For example, **FastTime**$_{11}$ refers to the eleventh value of **FastTime**. The directly read product input parameters are expressed in **boldface** to highlight dependencies. Parameters derived using those values (e.g. during a multi-looking process) are expressed in *italics*.

Table 4: ASAR Product Parameters Required for Geocoding (RD & GG)

Name used in this document	Section(s)	Table # *(see [3] & [8])*	Field Contents *(see [3] & [8])*	Field#	Data Type	Description
AzLineTime	MDS 1 - 5[1]	8.4.1.9.10-1	Zero Doppler Time in azimuth	1	`mjd`	Zero-Doppler time for an azimuth line [Modified Julian Day time]
AzSpacing	SPH	8.4.1.7-1	`AZIMUTH_SPACING`	30	`Afl`	Azimuth sample spacing [metres]
AzTimeFirst, AzTimeLast	MPP	8.4.1.9.2-1	First Zero Doppler Azimuth time, Last Zero Doppler Azimuth Time	1, 3	`mjd`	First (or Last) zero-Doppler azimuth line time [Modified Julian Day time]
DataOffset	SPH, DSD	**A**: 8.4.1.7-2 (non-WSS), or 8.4.1.7-3 (WSS); **B**: 5.4.2-1	DSD for MDS[1], DS_OFFSET	**A**: 45 (IM, WSM, GM1) 45-46 (AP), 42-46 (WSS); **B**: 4	`Ad`	Data offset within product [bytes]
FastTime	LADS	8.4.1.9.7-1	2-way slant range time to range sample	6, 9	`11*fl`	2-way slant range time at 11 range sample locations, for the first and last lines of the granule [seconds]
LineTimeInterval	SPH	8.4.1.7-1	`LINE_TIME_INTERVAL`	31	`Afl`	Azimuth sample spacing [seconds]
RadarFreq	MPP	8.4.1.9.2-1	Radar Frequency	44	`fl`	Radar carrier frequency [Hz]
RgSampleRate	MPP	8.4.1.9.2-1	Range Sampling Rate	43	`fl`	Range sampling rate [Hz]

[1] IMS, IMP, IMM, WSM, and GM1 products have one MDS; APS, APP, APM products have two (identical ZDT timelines); WSS products have five, one for each sub-swath (each with an individual ZDT timeline)

SwathHeight	SPH, DSD	**A**: 8.4.1.7-2 (non-WSS), or 8.4.1.7-3 (WSS); **B**: 5.4.2-1	DSD for MDS[1], NUM_DSR	**A**: 45 (IM, WSM, GM1) 45-46 (AP), 42-46 (WSS); **B**: 6	`Al`	Raster dataset height [samples]; for WSS mosaic "height" derivation, see [11]
SwathWidth	SPH, DSD	**A**: 8.4.1.7-2 (non-WSS), or 8.4.1.7-3 (WSS); **B**: 5.4.2-1	DSD for MDS[1], DSR_SIZE	**A**: 45 (IM, WSM, GM1) 45-46 (AP), 42-46 (WSS); **B**: 7	`Al`	Raster dataset width [samples] For non-WSS products: Width=(**DSR_SIZE**-RH[2])/ByteDepth For WSS mosaic "width" derivation, see [11]
ZDTF$_j$	LADS	8.4.1.9.7-1	Zero Doppler Time in azimuth of *first* line of the granule	1	`mjd`	Gives azimuth location of grid line for *first* line of the granule; j varies from 1 to N
FastTimeF$_{j,i}$	LADS	8.4.1.9.7-1	2 way slant range time to range sample for *first* line of the granule	6	`11*fl`	Two way slant range time [ns]; i varies from 1 to M=11
IncAngle$_{j,i}$	LADS	8.4.1.9.7-1	Incidence angle at range sample i	6	`11*fl`	*Nominal* incidence angle at each grid point (no DEM considered); i varies from 1 to M=11
ZDTL$_j$	LADS	8.4.1.9.7-1	Zero Doppler Time in azimuth of *last* line of the granule	8	`mjd`	Gives azimuth location of grid line for *last* line of the granule j varies from 1 to N
FastTimeL$_{j,i}$	LADS	8.4.1.9.7-1	2 way slant range time to range sample for *last* line of the granule	9	`11*fl`	Two way slant range time [ns]; i varies from 1 to M=11

[2] PF-ASAR image raster row header (RH) byte length is 17 bytes; Byte depth is 2 bytes for IMP/IMM/APP/APM/WSM/GM1, 4 bytes for IMS/APS/WSS

Table 5: ASAR State Vector Parameters used during Geocoding (RD)

Name used in this document	Section(s)	Table # *(see [3] & [8])*	Field Contents *(see [3] & [8])*	Field#	Data Type	Description
N_{SV}	MPP	8.4.1.9.2-1	Number of state vectors (5)	implicit	ul	Number of state vectors (5)
$S_t[j]$	MPP	8.4.1.9.2-1	Time of state vector	82	mjd	Time of state vector (converted to seconds w.r.t. azimuth start time)
$S_x[j]$	MPP	8.4.1.9.2-1	X position [converted from 10^{-2} m to m]	82	sl	X position in Earth fixed reference frame
$S_y[j]$	MPP	8.4.1.9.2-1	Y position [converted from 10^{-2} m to m]	82	sl	Y position in Earth fixed reference frame
$S_z[j]$	MPP	8.4.1.9.2-1	Z position [converted from 10^{-2} m to m]	82	sl	Z position in Earth fixed reference frame
$V_x[j]$	MPP	8.4.1.9.2-1	X velocity [converted from 10^{-5} m/s to m/s]	82	sl	X velocity in Earth fixed reference frame
$V_y[j]$	MPP	8.4.1.9.2-1	Y velocity [converted from 10^{-5} m/s to m/s]	82	sl	Y velocity in Earth fixed reference frame
$V_z[j]$	MPP	8.4.1.9.2-1	Z velocity [converted from 10^{-5} m/s to m/s]	82	sl	Z velocity in Earth fixed reference frame

Table 6: ASAR Slant/Ground Range Parameters used with IMP/IMM/APP/APM/WSM/GM1 Products (RD & GG)

Name used in this document	Section(s)	Table # *(see [3] & [8])*	Field Contents *(see [3] & [8])*	Field #	Data Type	Description
$ZDT_{SRGR,j}$	SR/GR ADSR	8.4.1.9.4-1	Zero Doppler Time in azimuth from which parameters apply	1	mjd	Azimuth ZDT for this update. j varies from 1…N_{SRGR}
$FastTime_{SRGR,j}$	SR/GR ADSR	8.4.1.9.4-1	2 way slant range time to first range sample	3	fl	Slant range distance (converted internally from ns to m)
$GR_{0,j}$	SR/GR ADSR	8.4.1.9.4-1	Ground range origin of the polynomial (GR_0) measured from the first pixel of the line	4	fl	Ground range origin
GTS_j	SR/GR ADSR	8.4.1.9.4-1	The coefficients $\mathbf{S_0}$, $\mathbf{S_1}$, $\mathbf{S_2}$, $\mathbf{S_3}$, and $\mathbf{S_4}$ of the ground range to slant range conversion polynomial. Slant range = $\mathbf{S_0} + \mathbf{S_1}(GR-GR_0) + \mathbf{S_2}(GR-GR_0)^2 + S_3(GR-GR_0)^3 + \mathbf{S_4}(GR-GR_0)^4$ where GR is the ground range distance from the first pixel of the range line	5	5*fl	Slant/Ground range coefficients to calculate a slant range distance given an image ground range position

Table 7: ASAR Geolocation Grid Parameters used for geolocation-grid (GG) based geocoding

Name used in this document	Section(s)	Table # *(see [3] & [8])*	Field Contents *(see [3] & [8])*	Field #	Data Type	Description
$\mathbf{LatF_{j,i}}$	LADS	8.4.1.9.7-1	geodetic latitude (positive north)	6	`11*sl`	latitude of grid points for *first* line of granule [degrees][3]; i varies from 1 to M=11
$\mathbf{LonF_{j,i}}$	LADS	8.4.1.9.7-1	geodetic longitude (positive east)	6	`11*sl`	longitude of grid points for *first* line of granule [degrees][3]; i varies from 1 to M=11
$\mathbf{LatL_{j,i}}$	LADS	8.4.1.9.7-1	geodetic latitude (positive north)	9	`11*sl`	latitude of grid points for *last* line of granule [degrees][3]; i varies from 1 to M=11
$\mathbf{LonL_{j,i}}$	LADS	8.4.1.9.7-1	geodetic longitude (positive east)	9	`11*sl`	longitude of grid points for *last* line of granule [degrees][3]; i varies from 1 to M=11

[3] Converted from 10^{-6} deg. to be stored internally as deg

3 SYSTEM DESIGN

The processing steps inherent in any geocoding system for PF-ASAR products may be broadly arranged as consisting of the following steps:

- PF-ASAR Product Ingestion
  - Detection
  - Debursting (*for WSS*)
  - Multi-looking
- DEM Input & Map Geometry Initialisation
- Radar Geometry Initialisation
  - Product or External State Vector Format Input
  - Timing Annotations Initialisation
- DEM Traversal
  - Point Geolocation
    - Coordinate Transformation
- Annotation Output

The major system design features of an ASAR geocoding software system are shown in Figure 1. A brief summary is provided below; further details are presented in later sections.

The relevant product annotations are first read and a contiguous 2-D radar image (or images in the AP case) is loaded from the input ASAR product. If the product measurements are *complex* (IMS/APS/WSS), the values are "detected" to first generate radar intensity values; WSS products are also "debursted" and "mosaiced" into a single contiguous slant range image [12]. Within the input stage, the product may also be multi-looked if necessary to a resolution compatible with the output geocoded image resolution that is desired.

Once the product has been ingested, the raster geocoding algorithm can begin. The multi-looked radar geometry image is read into memory, and the geographic area of interest is traversed progressively. Either a DEM or a simpler ellipsoid model presented in map geometry is used to describe the position of each point within the area: the 2D cartographic (northing, easting) or geographic (latitude, longitude) positions are transformed into Cartesian (x,y,z) coordinates. For each point, the ENVISAT satellite modelled state vectors (available also in Cartesian coordinates) are used to determine the image's azimuth row corresponding to Zero-Doppler time, the azimuth convention used by the standard PF-ASAR processor. After a small azimuth timing correction (described in detail later) is applied to consider satellite movement between pulse transmission and reception, the slant range may also be calculated (and converted to ground range if necessary). Now that both the azimuth and range coordinates corresponding to the current map position are known, the image content can be resampled from the known location in radar geometry into the current map geometry position. One traverses the 2D map geometry grid, geolocating in the manner described, until all DEM grid points have been visited and the geocoding is complete.

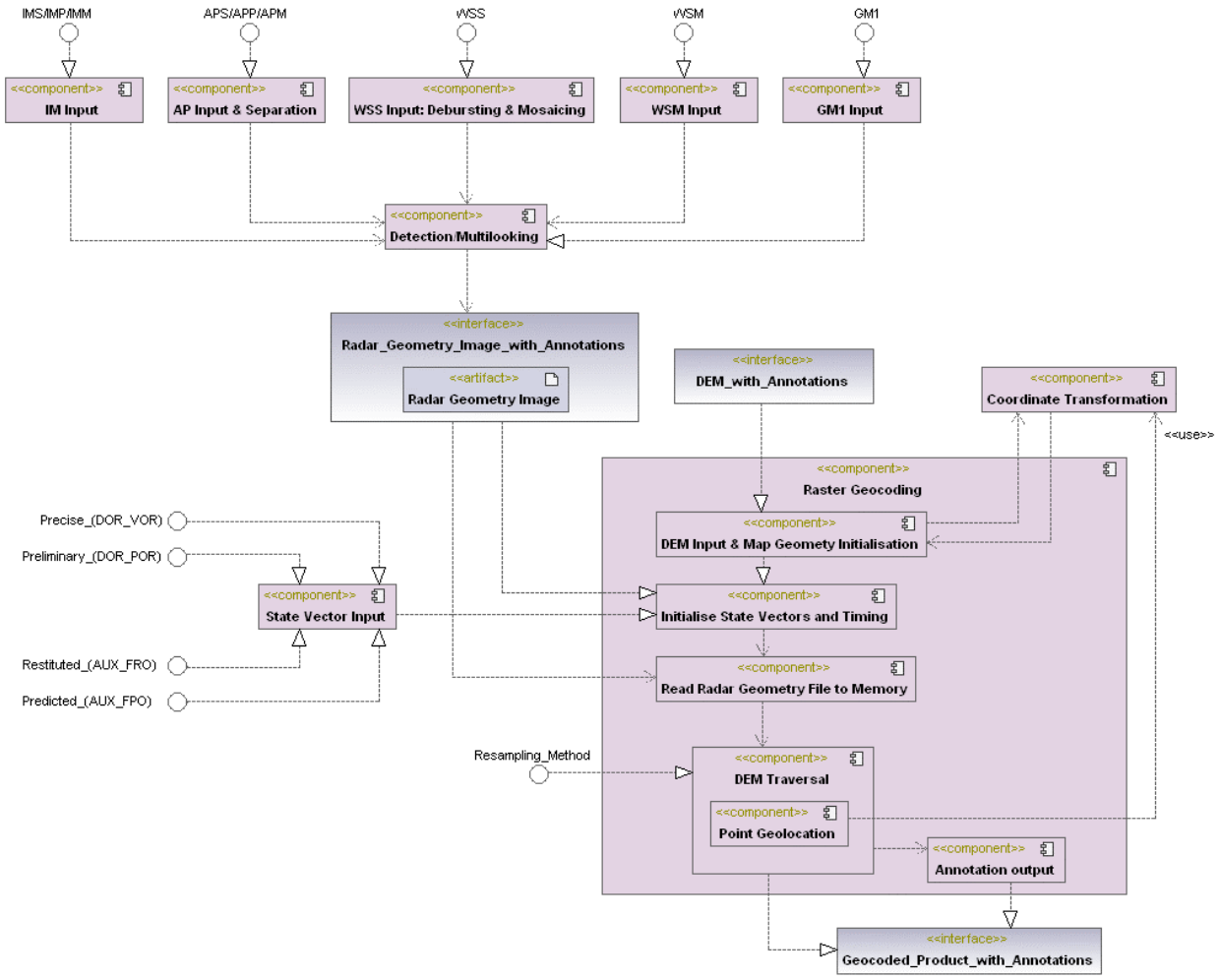


Figure 1: SAR Geocoding System Components

4 SAR GEOCODING IN DETAIL

The components of the ASAR geocoding algorithm (see Figure 1) are described in the following sections.

4.1 Ingestion to Annotated RADAR Geometry Raster

The tasks accomplished during the initialisation phase are described in the following subsections.

4.1.1 Product Input

4.1.1.1 IMS/IMP/IMM Product Input

In the case of ASAR images acquired in *Image Mode* (**IM**), there is a single focussed contiguous image that is input to the geocoder. Depending on product type, that image is presented either in slant range (IMS), a single contiguous ground range geometry (IMP), or a sliding ground range geometry with multiple slant/ground range conversion updates provided along the azimuth dimension (IMM).

While IMP and IMM products present themselves as scalar images with nominally "square" pixels on the ground, an IMS product provides complex (real & imaginary components) of the radar backscatter with the (spatial ground-equivalent) azimuth sampling interval significantly shorter than the equivalent ground range sampling interval. One may therefore choose to first carry out a detection operation, retrieving radar backscatter amplitude from the IMS "single-look-complex" (SLC) complex image content and averaging over a defined window to generate an image with nominally "square" pixel sizes that is then input to the geocoder. The detection and multi-looking step is described in section 4.1.2.

4.1.1.2 APS/APP/APM Product Input and Separation

In the case of ASAR images acquired in *Alternating Polarisation* Mode (**AP**), there are two focussed contiguous images that are input to the geocoder. Depending on product type, those two images are presented either in slant range (APS), a single contiguous ground range geometry (APP), or a sliding ground range geometry with multiple slant/ground range conversion updates provided along the azimuth dimension (APM).

All AP products contain not one MDS (as is true for all IM products), but two: MDS1 & MDS2. For all AP products, each of the steps described previously (identical in nature to those listed for IM products above) must therefore be carried out not on a single raster, but on two. It is advisable to *separate* each MDS into distinct raster files that may later each be input to the geocoding algorithm independently if desired.

While APP and APM products present themselves as scalar images with nominally "square" pixels on the ground, an APS product provides complex (real & imaginary components) of the radar backscatter with the (spatial ground-equivalent) azimuth sampling interval significantly shorter than the equivalent ground range sampling interval. One may therefore choose to first carry out a detection operation, retrieving radar backscatter amplitude from the "single-look-complex" (SLC) complex samples within the APS product, and averaging over a defined window to generate an image with nominally "square" pixel sizes that is then input to the geocoder. The detection and multi-looking step is described in section 4.1.2.

Important Note: AP products processed with versions of the PF-ASAR prior to v4.02 are subject to a product *annotation timing bias* than can be corrected during this input stage [1]. Failing to perform the corrections on these products can introduce significant absolute location errors later.

4.1.1.3 WSM/GM1 Product Input

In the case of ASAR images acquired in *Wide Swath* (**WS**) mode processed into WSM products, the five sub-beams SS1 to SS5 are stitched together into a single detected focussed contiguous image that can be input to the geocoder. The same is true (at a coarser resolution) for *Global Monitoring* (**GM**) images processed into GM1 products, and (at a finer resolution) for narrower swath IMM and APM products.

The images are presented in a sliding ground range geometry with multiple slant/ground range conversion updates provided along the azimuth dimension. The parameters necessary to describe this kind of geometry are listed in Table 6.

4.1.1.4 WSS Product Input: Debursting and Mosaicing

The ASAR WSS product is a special case among PF-ASAR products: both debursting and mosaicing procedures [11][12] are recommended as pre-processing steps before geocoding is performed. The pre-processing should produce a single contiguous slant range image arranged in a raster format [11]. That may then be processed during geocoding as a simple slant range image (without any slant/ground range considerations).

4.1.1.5 Ingested Parameters for Geocoding

Note that although knowledge of the *spatial* range and azimuth pixel spacings is important, it is important to emphasise that imaging radars actually measure *time* differences between transmission and reception of echoes. Also in the azimuth direction, it is *time*-tagging of the radar measurements that is the primary reference. *Spatial* state vector positions are attached to those same times later as secondary (nonetheless very important) items of information. Yet it cannot be overstated that geolocation of radar images is tied to timing information in a way that optical imagery is not: for radar images, the primacy of timing annotations extends also into the cross-track dimension. Radar ranging measurements are primarily quantifications of time separation, and only secondarily measurements of distance.

Summarising the above sections, the parameters required for each product type are listed in Table 8. Note that the first rows contain the primary geometry information, namely the radar timing annotations that are necessary to geocode an image. In the case of WSS products, it is assumed that it is a (derivative) *mosaic* that is being geocoded.

Table 8: Geometry parameters required for geocoding different ASAR product types (RD: required for Range-Doppler geocoding, GG: required for LADS-based geocoding)

	Parameter	Product Type								
		IMS	APS	WSS	IMP	APP	IMM	APM	WSM	GM1
Radar Timing Annotations	Azimuth Start Time **AzTimeFirst** / $AzTimeFirst_{ML}$	✔			✔		✔			
	Azimuth Stop Time **AzTimeLast** / $AzTimeLast_{ML}$	✘ *Redundant*: can be derived as $\mathbf{AzTimeFirst} + (\mathbf{SwathHeight} - 1) \cdot \mathbf{LineTimeInterval}$								
	Slant range (near) $\mathbf{FastTime_1}$ / $FastTimeNear_{ML}$	✔			✔		✔			
	Slant range (far) $\mathbf{FastTime_{11}}$ / $FastTimeFar_{ML}$	✘ *Redundant*: can be derived as $\mathbf{FastTime_{11}} = \mathbf{FastTime_1} + (\mathbf{SwathWidth} - 1) \cdot \frac{c}{2 \cdot \mathbf{RgSampleRate}}$								
	Azimuth sample interval **LineTimeInterval** / $LineTimeInterval_{ML}$	✔			✔		✔			
	Range sample interval **RgSampleRate** // $RgSpacing_{ML}$	✔			✔		✔			
Image Size	Image raster width **SwathWidth** / *Width*	✔			✔		✔			
	Image raster height **SwathHeight** / *Height*	✔			✔		✔			

	Parameter	Product Type								
		IMS	APS	WSS	IMP	APP	IMM	APM	WSM	GM1
State Vectors	One of product header state vectors or same from external source (DOR_VOR, DOR_POR, AUX_FRO, AUX_FPO) **N_{SV} , $S_t[1…N_{SV}]$, $S_x[1…N_{SV}]$, $S_y[1…N_{SV}]$, $S_z[1…N_{SV}]$**	RD			RD		RD			
Slant/ground range conversion parameters	Number of slant/ground range polynomials **N_{SRGR}**	-	-	-	1	1	N_{SRGR}	N_{SRGR}	N_{SRGR}	N_{SRGR}
	Ground range origin **$GR_{0,j}$**	-			✔		✔			
	Slant/ground range polynomial coefficients **GTS_j**	-			✔		✔			
	Azimuth time reference for each set of coefficients **$ZDT_{SRGR,j}$**	-			-		✔			
Geolocation Grid Parameters	Latitudes in *first* line of granule **$LatF_{j,i}$**	GG		Inadvisable due to disjoint sets of LADS records	GG		GG			
	Longitudes in *first* line of granule **$LonF_{j,i}$**									
	Zero-Doppler Time of *first* line of granule **$ZDTF_j$**									
	Latitudes in *last* line of granule **$LatL_{j,i}$**									
	Longitudes in *last* line of granule **$LonL_{j,i}$**									
	Zero-Doppler Time of *last* line of granule **$ZDTL_j$**									

4.1.2 Detection and Multi-looking

Before geocoding, if the input product contains complex values (IMS, APS, WSS), they may be "detected" to radar backscatter intensity values. Next, those values may optionally be multi-looked in one or both of the range and azimuth dimensions to generate a radar geometry image with approximately "square" sample sizes.

First, the complex radar signal is converted to scalar intensity values by taking the sum of the squares of the real and imaginary parts; this is called *detection*. The intensity for a given cell, referred to in [10] as $DN_{i,j}^2$ for input position (i, j) is:

$$Intensity \equiv DN_{i,j}^2 = (SLC_{real})^2 + (SLC_{imag})^2 \quad \textbf{(1)}$$

where

$DN_{i,j}$ = input digital number

SLC_{real} and SLC_{imag} = real and imaginary components of the complex input value.

To by default generate a multi-look output image with approximately square sample dimensions at mid-range, the following method is applied.

The mid-range distance is retrieved from the **FastTime** input array as:

$$\text{FastTime}_{\text{mid}} \leftarrow [\textbf{\textit{FastTime}}_1 + \textbf{\textit{FastTime}}_{11}] / 2$$

The mid-range incidence angle is retrieved as:

$$\text{IncAngle}_{\text{mid}} \leftarrow \textbf{\textit{IncAngle}}_6$$

Although the timing annotations have primacy (as discussed above), it is also useful to have the *spatial* equivalents. The two-way slant range sampling rate in *time* is converted to one-way slant range *distance* via:

$$RgSpacing = \frac{c}{2 \cdot \textbf{RgSampleRate}} \quad \textbf{(2)}$$

Default range and azimuth multi-looking factors can be calculated to satisfy the following relation:

$$\frac{nAzLooks}{nRgLooks} = \frac{RgSpacing}{\textbf{AzSpacing} \cdot \sin(IncAngle_{mid})} \quad \textbf{(3)}$$

If one of the multi-looking factors is specified manually, then the other can be calculated using the above relation and then rounded to the nearest integer. When factors are applied that satisfy the above relation, one produces samples that are as "square" as possible in ground range geometry.

4.1.2.1 Modified multi-look sample intervals

The timing interval between successive azimuth lines is of primary importance in geolocation. The multi-looked interval is derived simply as:

$$LineTimeInterval_{ML} = \textbf{LineTimeInterval} \cdot nAzLooks \quad \textbf{(4)}$$

The spatial multi-look sample intervals are derived from the input spacings simply as:

$$RgSpacing_{ML} = RgSpacing \cdot nRgLooks \quad \textbf{(5)}$$

$$AzSpacing_{ML} = \textbf{AzSpacing} \cdot nAzLooks \quad \textbf{(6)}$$

4.1.2.2 Calculation of multi-look near-range and azimuth-start times

Given a sample-centred annotation convention (generally used within this document unless otherwise noted), the equations relating the original product's annotated values and the new, adjusted multi-looked values are:

$$AzTimeFirst_{ML} \leftarrow \textbf{AzTimeFirst} + \left[\textbf{LineTimeInterval} \cdot \left(\frac{nAzLooks - 1}{2}\right)\right] \quad \textbf{(7)}$$

The eleven 2-way slant range time values are extracted from field 6 of the LADS (see Table 5). The first value is used as the near range, the eleventh value as the far range.

The multi-look near range fast time value is obtained from the near-range (first) value of **FastTime**:

$$FastTimeNear_{ML} \leftarrow \mathbf{FastTime}_1 + \left[RgSpacing \cdot \left(\frac{nRgLooks - 1}{c} \right) \right] \quad \textbf{(8)}$$

where c is the speed of light.

4.1.2.3 Modified Multi-look Boundaries

Image near and far range values are referred to as r_{near} and r_{far}. They are obtained from the near-range value $FastTimeNear_{ML}$, the image sample dimensions and sample intervals, as:

$$r_{near} = FastTimeNear_{ML} \cdot \frac{c}{2} \quad \textbf{(9)}$$

where c is the speed of light.

For slant range images, the image width (number of slant range samples) is known to be:

$$Width_{SR} = \frac{\left(r_{far} - r_{near} \right)}{\left(RgSpacing \cdot nRgLooks \right)} + 1 \quad \textbf{(10)}$$

Alternatively:

$$r_{far} = r_{near} + (Width_{SR} - 1) \bullet RgSpacing_{ML,} \quad \textbf{(11)}$$

where $Width_{SR}$ is the number of range samples in a slant range image.

The image height $Height_{az}$ (azimuth extent in samples) should be consistent with the other multi-look boundary information:

$$Height_{az} = \frac{\left(AzTimeLast_{ML} - AzTimeFirst_{ML} \right)}{\left(LineTimeInterval_{ML} \right)} + 1 = \frac{\mathbf{SwathHeight}}{nAzLooks} \quad \textbf{(12)}$$

4.2 DEM Input and Map Geometry Initialisation

The interface to the reference DEM that is to be used must ensure that all required cartographic and geodetic parameters annotating the DEM are available during geocoding. These parameters (e.g. false easting, false northing, central meridian, standard parallel(s), datum shift) must be sufficient to specify the conversion of a point coordinate from cartographic (northing, easting, height) coordinates (P_E, P_N, P_h) or geographic coordinates (P_λ, P_ϕ, P_h) to first locally, and then globally-referenced Cartesian values (P_x, P_y, P_z). The different coordinate systems are summarised in Table 9.

Cartographic coordinates are generally parameterised using a so-called “mapset” in a manner conforming to a small number of map projection types, such as Transverse Mercator (TM), Oblique Mercator (OM), Polar Stereographic (PS), Oblique Stereographic (OS), Lambert Conformal Conic (LCC) etc. – each maps a 3D position on the Earth into a defined 2D map projection. Each mapset also specifies a given *local ellipsoid* (semi-major and semi-minor axis lengths) as well as a seven-parameter “datum shift” (including 3D vector translation and rotation vectors as well as a scalar scale factor) that detail how to transform 3D Cartesian coordinates from that local ellipsoid into a *global geodetic ellipsoid reference* such as WGS84 or a reference frame such as ITRF2005. The geocoding software must be able to deal with a variety of map projection types and instances (typically through the use of a software library) to enable access to DEM rasters presented in a variety of projections, ellipsoids, and datum shifts.

Table 9: Cartographic & Geodetic Coordinate Systems

	Coordinate System			
	Map Projection	**Geographic**	**Cartesian (based on *local* Ellipsoid)**	**Cartesian (based on *global* Ellipsoid)**
Axes	Easting E	Longitude λ	x'	x
	Northing N	Latitude ϕ	y'	y
	Height h	Height h	z'	z
Point	(P_E, P_N, P_h)	(P_λ, P_ϕ, P_h)	$(P_{x'}, P_{y'}, P_{z'})$	(P_x, P_y, P_z)
Examples	Swiss Oblique Mercator	Swiss lat/long (Bessel ellipsoid)	Bessel Cartesian	WGS84
	UTM Zone 32	WGS84 lat/long	WGS84 (no datum shift: local=global)	WGS84

Global ellipsoid-based Cartesian coordinates can be expressed either in Earth-Centred-Rotating (ECR) convention, or Earth-Centred-Inertial (ECI). Inertial is meant here in the sense that the frame of reference is not subject to accelerations (e.g. rotation): its frame is fixed with respect to the stars. The Cartesian coordinates of a point (P_x, P_y, P_z) derived from a DEM position are expressed more easily in the same Earth-Centred-Rotating (ECR) coordinate system used also for satellite state vectors.

Given that both are expressed in the same ECR reference frame, it is relatively straightforward to use (P_x, P_y, P_z) during geolocation to determine the position of the satellite (S_x, S_y, S_z) corresponding to the Doppler annotation convention of the product. The details of that calculation are provided in section 4.6.

4.3 State Vectors

No state vectors are technically *required* for geolocation if the product's annotated geolocation grid (GG method, see section 4.6.1) is used for geocoding. However, to achieve the highest geolocation accuracy possible, one must employ range-Doppler geocoding: in that case, a set of highly accurate state vectors is required to enable straightforward tiepoint-free geolocation.

4.3.1 State Vector Input

Given a range-Doppler geocoding approach, the algorithm requires either satellite state vectors from the ASAR product annotations (see Table 5), or alternatively, values read from an external ENVISAT state vector product. The source (and therefore quality) of the state vectors included in the product annotations is itself annotated in the product. Validation experience has shown that restituted, preliminary, and precise state vectors generally produce acceptable geolocation results.

The four different external state vector product types are summarised in Table 10, in ascending qualitative order.

Table 10: External ENVISAT State Vector Product Types

Name	**File Prefix**	**Description**
Predicted	AUX_**FPO**_AX	Flight-segment Predicted Orbit (lowest quality)
Restituted	AUX_**FRO**_AX	Flight-segment Restituted Orbit (usually included in ASAR products)
Preliminary	DOR_**POR**_AX	DORIS preliminary, typically available days after acquisition
Precise	DOR_**VOR**_AX	DORIS precise, typically available one month after acquisition

The ASAR product annotations generally provide state vectors at times distributed approximately between the image's start and stop times. However, the external state vectors provide more sparsely sampled positions and velocity estimates that must be interpolated to provide more scene-specific information.

4.3.2 State Vector Initialisation and Timing

To expedite processing later, one can define a look-up table to hold the satellite position (S_x, S_y, S_z) and velocity (V_x, V_y, V_z) corresponding to each azimuth line (t=**AzLineTime**). The positions must be interpolated from the sparser product annotation values or the even less dense external state vector file positions. Performing the interpolation once during this initialisation stage speeds up the geocoding process later. B-Spline, Hermitian, and Lagrange polynomial based interpolation schemes [5] [6] [17] (as well as the ENVISAT *CFI* software `ppf_orbit` [2]) use available position and velocity data points to model (S_x, S_y, S_z) and (V_x, V_y, V_z) at instances in time *between* the available state vectors. Each method has yielded acceptable results; in non-sparse short-arc cases, a simple polynomial fit (without consideration of the velocities) can also suffice.

The time corresponding to each azimuth line ($\forall$j: 0…$Height_{az}$ -1) is calculated as:

$$t_j = AzLineTime = AzTimeFirst_{ML} + (j-1)\cdot LineTimeInterval_{ML}, \quad \textbf{(13)}$$

where j denotes the image line in question. When $L_{az} = Height_{az} - 1$ is the last line in the input image, with j varying from 0 to $L_{az,}$, the look-up table is constructed as follows:

$$\vec{S}_0 = \left(S_x(t_0), S_y(t_0), S_z(t_0)\right)\ldots \quad \vec{S}_j = \left(S_x(t_j), S_y(t_j), S_z(t_j)\right)\ldots \quad \vec{S}_{L_{az}} = \left(S_x(t_{L_{az}}), S_y(t_{L_{az}}), S_z(t_{L_{az}})\right) \quad \textbf{(14)}$$

The values in this look-up table are used later during raster geocoding. Rather than recalculating them repeatedly within a raster geocoding algorithm, tabulating them as a look-up table during an initialisation step can speed up the geolocations performed later.

4.4 RADAR Geometry Image Input

The input file annotations are ingested, memory is allocated to hold the binary image, and the input dataset is read into the array. Implementing the geocoder to ingest the whole of the radar geometry image into main memory increases the requirement on memory by a few hundreds of megabytes, but this is easily accommodated on modern computing platforms. Keeping the radar geometry raster in memory ensures that the algorithm remains insensitive to changes in the relative orientation between the azimuth (path) axis and the northing axis of the reference DEM. Depending on the map projection selected and location under investigation on the Earth, the relative orientations of these two axes can vary from being nearly parallel (e.g. UTM near equator) to oblique or perpendicular (e.g. polar stereographic projections near the poles).

4.5 Orthorectification / DEM Traversal

DEM traversal refers to two loops within the geocoding algorithm, moving between a minimum and maximum northing (or latitude) coordinate, and at each of those DEM "lines" between a minimum and maximum easting (or longitude) coordinate "column". The DEM boundaries are generally either (a) set by the user at run-time, or (b) set to "enclosing" values determined from the scene's corner latitude and longitude corner coordinates. The latitude and longitude of the corners (provided by fields 7 to 18 of the SPH detailed in Table 8.4.1.7-1 in [3]) can be converted into the DEM's cartographic reference system to provide useful default values. That yields minimum and maximum northing values N_0 and N_1, as well as minimum and maximum Easting values E_0 and E_1. Their values should be *regularised* to be exact integer sample multiples of the DEM's reference sample centres. One can loop within the subset of a DEM file delineated by those values and output the resampled SAR backscatter measurements in the DEM's map geometry.

One can choose the proper order of DEM traversal to allow simultaneous computation of the occurrence of local layover or radar shadow [7], but that is beyond the scope of this document.

Given a DEM defined between E_{D0} and E_{D1} in easting and N_{D0} and N_{D1} in northing, with sample intervals of ΔN (northing) and ΔE (easting), the DEM raster index of the point (E, N) is for easting:

$$I_E = (E - E_{D0}) / \Delta E \quad \textbf{(15)}$$

For the typical *maximum northing at beginning of file* convention, the northing DEM raster index is:

$$I_N = (N_{D1} - N) / \Delta N \quad \textbf{(16)}$$

The number of samples (width) of the output GTC image generated for the box delineated between N_0 and N_1, and E_0 and E_1 is:

$$Width_{GTC} = (E_1 - E_0) / \Delta E + 1 \quad \textbf{(17)}$$

Similarly in the northing dimension:

$$Height_{GTC} = (N_1 - N_0) / \Delta N + 1 \quad \textbf{(18)}$$

The DEM resolution in planimetry, as specified by ΔE and ΔN should be greater than or equal to the nominal ground resolution of the multi-looked image being geocoded. Complying with that requirement ensures that loss of image fidelity through the geocoding/resampling process is kept to a minimum. If only a well-resolved DEM is available, then depending on the user's needs, either (a) the DEM should be oversampled to increase the ground sampling rate (of course the *resolution* remains unaltered), or (b) the input radar image should be multi-looked to a resolution more compatible with the DEM available. Choice (a) is generally implemented as a pre-processing step, as that type of processing falls outside the primary scope of a SAR geocoding software package. Choice (b) can be appropriate if the user is interested in using multi-looking to reduce the noise present in the radar image content or cutting the required processing time.

In an orthorectification procedure that employs "*backward* geocoding", the range and azimuth image indices are retrieved for each DEM grid point (northing, easting). The reverse procedure "*forward* geocoding" takes a slant range and azimuth time as input and uses the same geolocation equations to determine a map geometry (northing, easting, height) coordinate that is compatible with the input radar geometry coordinates. Image product geocoding is best done using a "backward" technique – the remainder of this section therefore concentrates exclusively on the "backward geocoding" methodology.

After geolocation, once the range and azimuth indices are determined, a value is extracted from the input image content using the user-selected resampling kernel, and output in the DEM's map geometry. The process of geolocation followed by resampling is illustrated in Figure 2. The details of how the range and azimuth raster indices are retrieved, and how those are used during resampling are discussed in the following sections. The next section gives a broad overview of the orthorectification procedure by reviewing the "backward geocoding" algorithm used in image geocoding, whereby the DEM is traversed along easting and northing axes.

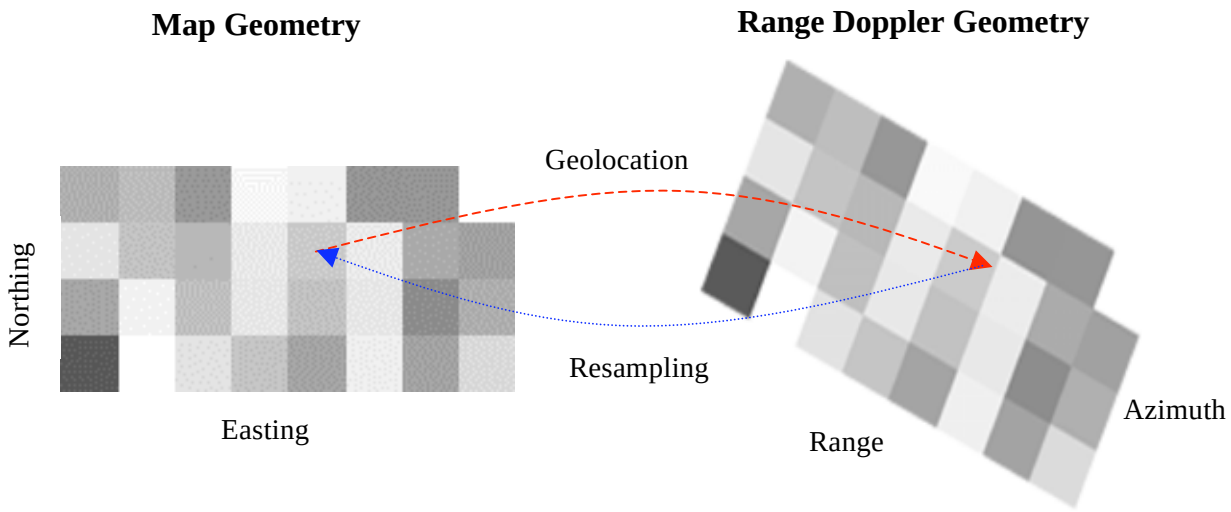


Figure 2: SAR Image Orthorectification via Backward Geocoding

The orthorectification procedure (DEM traversal) proceeds as shown in the following pseudocode:

Initialise azimuth time look-up table S_t = *AzLineTime*[0…*Height*$_{az}$-1]

Read available state vector information (S_x,S_y,S_z) & (V_x,V_y,V_z)

If (RD geolocation active) then

Initialise azimuth satellite position and velocity look-up tables: interpolate (S_x,S_y,S_z) & (V_x,V_y,V_z) to generate *ENVISAT_pos*[0…*Height*$_{az}$-1] and *ENVISAT_vel*[0…*Height*$_{az}$-1]

Endif

If (GG geolocation active) then

Initialise buffer holding LADS (geolocation grid)

Endif

for (N = N_1 to N_0 step -ΔN) Cycle through DEM *rows* of interest

Update raster index I_N

Initialise line buffer to hold values for DEM positions from E_0 to E_1

for (E = E_0 to E_1 step ΔE Cycle through DEM *columns* of interest

Update raster index I_E

Retrieve local height h from DEM at index position (I_E, I_N)

Convert map coordinate (P_E, P_N, P_h) to global geographic $P_{\lambda,\phi,h}$ system

Convert map coordinate (P_E, P_N, P_h) to global Cartesian (P_x, P_y, P_z) system

Point geolocation (see devoted section for more detail):

If (RD geolocation active) then — RD: Range Doppler-based geolocation

Determine spacecraft position within *ENVISAT_pos LUT* fulfilling Doppler condition with point (P_x, P_y, P_z)

Endif

If (GG geolocation active) then — GG: Geolocation Grid-based geolocation

Interpolate grid to determine range and azimuth line time indices I_r & I_a corresponding to (P_λ, P_ϕ, P_h)

Endif

Resample: retrieve radar geometry image content σ at coordinates (I_r, I_a)

Place retrieved content σ in current line buffer using index (I_E, I_N)

end for — Move on to next *easting* value

Write line buffer for current DEM line

end for — Move on to next *northing* value

Write header information annotating complete GTC image produced

4.6 Point Geolocation

This section describes geolocation algorithms for a single point (P_x ,P_y ,P_z), which could be either one of many within a greater DEM traversal algorithm (described in the previous section), or just a single point (e.g. corner reflector) being used to validate a product's geometry. Two geolocation algorithms are described here, labelled (a) "geolocation grid", and (b) "range-Doppler". The goal of both algorithms is to retrieve the image indices within the PF-ASAR product under study (I_a and I_r) that correspond to the input point (P_x, P_y, P_z) known to exist on the Earth's surface. If the point is being geocoded within a DEM traversal, then its DEM index coordinates (I_N and I_E) are already known: I_a and I_r must now be obtained.

The localisation improvement and resampling issues that follow once those coordinates are known do not substantially differ between the GG & RD methods – they are treated in later sections.

4.6.1 Geolocation Grid (GG)

This simple interpolation method uses the latitude/longitude geolocation grid (LADS record) available in all radar geometry PF-ASAR products. Height variations within a scene are not considered in this algorithm. Note that the geolocation grid is computed during product generation using the state vectors that are *available at the time of production* of the PF-ASAR product. The geolocation grid is in a sense "*burned in*" using the information available at the time. If more precise state vectors become available *after* the product is generated, they cannot easily be used to improve the accuracy of the geolocation, as state vector information is

ignored by GG-based geocoding. Due to the "burn in" and height variation issues, it is generally recommended that users if at all possible instead use the range-Doppler algorithm described in the next section.

Still, some users of imagery covering flat areas (e.g. ocean-marine environment) may elect to use this algorithm due to its relative simplicity.

The algorithm proceeds as follows:

Read complete LADS record (geolocation grid) into memory (number of azimuth updates x 11 range values) with geodetic latitude & longitude as well as slant range and azimuth zero-Doppler time

For the point of interest (P_x ,P_y ,P_z), obtain the equivalent expression in global geographic coordinates (P_λ ,P_ϕ ,P_h).

Traverse the cells of the geolocation grid linearly and use a point-in-polygon test to determine the cell coordinates that are immediately adjacent and enclose the point.

Perform biquadratic fits to geolocation grid values, modelling dependence of (a) slant range on latitude and longitude, and (b) azimuth ZDT on the same

Retrieve estimate of slant range R given coordinate (P_λ ,P_ϕ ,P_h) using biquadratic model

Retrieve estimate of azimuth time T given coordinate (P_λ ,P_ϕ ,P_h) using biquadratic model

Convert slant range (fast time) and azimuth ZDT (slow time) estimates into raster image indices

Use knowledge of slant range to determine local bistatic bias – calculate corrected azimuth time $T_c = T + R*2/c$

If the product in question is in a ground range projection, transform the estimated slant range R to ground range G

Use the azimuth time T_c to retrieve the azimuth image index value I_a

Use the slant or ground range distance (as appropriate) to retrieve the range image index I_r

Return the obtained values I_a and I_r

4.6.2 Range-Doppler (RD)

The most accurate geolocation algorithm solves the Doppler and range equations to determine the range and azimuth positions corresponding to the position (P_x ,P_y ,P_z) on the Earth.

Range Doppler geocoding for ASAR proceeds as follows:

Read state vector information (S_x,S_y,S_z) & (V_x,V_y,V_z) in ECR global Cartesian reference

Initialise azimuth satellite position and velocity look-up tables for PF-ASAR product: interpolate (S_x,S_y,S_z) & (V_x,V_y,V_z) to generate *ENVISAT_pos*[0…$Height_{az}$-1] and *ENVISAT_vel*[0…$Height_{az}$-1]

Read Earth terrain location: point (P_x,P_y,P_z) in global Cartesian reference

Determine the azimuth time T of an ENVISAT position corresponding to the Zero-Doppler Condition

Given T: now calculate the slant range distance as $R(T) = |\boldsymbol{S}(T) - \boldsymbol{P}|$

Use knowledge of slant range to determine local bistatic bias – calculate corrected azimuth time $T_c = T + R(T)*2/c$

Given T_c: calculate updated the slant range distance as $R(T_c) = |\boldsymbol{S}(T_c) - \boldsymbol{P}|$

If the product in question is in a ground range projection

Transform the slant range R to ground range G

Endif

Use the azimuth time T_c to retrieve the azimuth image index value I_a

Use the slant or ground range distance (as appropriate) to retrieve the range image index I_r

Return the obtained values I_a and I_r

Individual steps within the algorithm listed above are detailed in the following subsections.

4.6.2.1 Azimuth Timing: Azimuth Index Computation

For a point P on the Earth's surface, the following steps are performed:

If not already available, the native map geometry position (P_E, P_N, P_h) is converted into global Cartesian coordinates (P_x, P_y, P_z).

The orbit-position look-up table is searched (iteratively [7] or via bisection [9]), beginning at the most recent position, and the Doppler frequency corresponding to each position is calculated until a value greater than the expected zero reference is found (illustrated in Figure 3 by the red and blue dotted lines). As the search through positions S_{-3}, S_{-2}, S_{-1}, and S_{+1} advances, two values are always retained in memory: the last Doppler that was lower than zero (S_{-1}), and the first one found to be greater (S_{+1},).

The Doppler frequency is a function of the sensor position S for a given Earth position (P_x, P_y, P_z):

$$f_D(t) = -\frac{2}{\lambda} \cdot \left(\overrightarrow{v_P(t)} - \overrightarrow{v_S}\right) \cdot \left(\frac{\vec{P} - \overrightarrow{S(t)}}{\left|\vec{P} - \overrightarrow{S(t)}\right|} \right) \qquad \textbf{(19)}$$

where

$f_D(t)$ = Doppler frequency at time t

λ = wavelength of radar carrier frequency

$\overrightarrow{v_S(t)}$ = velocity of the sensor at time t

$\overrightarrow{v_P}$ = velocity of the Earth position P

$\vec{P}$ = (P_x, P_y, P_z), the position on the Earth

$\overrightarrow{S(t)}$ = $\left(S_x(t), \quad S_y(t), \quad S_z(t)\right)$, the position of the sensor at time t

t = azimuth (slow) time of sensor position

The velocity of P depends on its location on the surface of the Earth, which has a rotational velocity $\overrightarrow{\omega_E}$:

$$\overrightarrow{v_p} = \overrightarrow{\omega_E} \times \vec{P} \quad \textbf{(20)}$$

Since the spacecraft positions are defined for an Earth-centred rotating (ECR) coordinate system where $\overrightarrow{\omega_E} \equiv 0$, we have

$$\overrightarrow{v_p} = 0 \quad \textbf{(21)}$$

The problem at hand is the determination of the appropriate azimuth time that results in a zero Doppler value.

For single points, a bisection algorithm such as Brent's method [9] is an appropriate solution. In a raster geocoding context with many repeated similar but slightly varying computations, search through a look-up table may be more efficient. The following algorithm implements such a search:

```
(Initialise spacecraft position look-up table if not already done in a previous step)

Read Earth surface location (Px,Py,Pz) to be geolocated

Initialise azimuth LUT index j to zero (or keep at previous value if already set in a prior run)

NegDoppFound ← FALSE

PosDoppFound ← FALSE

while (!NegDoppFound and !PosDoppFound and j<jMax)

        Calculate fD between (Px,Py,Pz) and spacecraft position (Sx,Sy,Sz): S(j)

        If (fD < 0) then

                NegDoppFound ← TRUE

        Endif

        If (fD ≥ 0) then

                PosDoppFound ← TRUE

        Endif

        Increment azimuth index:  j ← j + 1

endwhile
```

Note relevant indices:

AzZDT_neg ← *j*-1

AzZDT_pos ← *j*

Use linear interpolation to determine fractional index value f_j corresponding to actual $f_D=0$

Return fractional index value f_j describing azimuth index and ZDT time when point ***P*** was imaged

Linear interpolation between the Doppler values at S_{-1} and S_{+1} yields an estimate of the sensor position $\vec{S}$ (S_x, S_y, S_z) corresponding to zero Doppler. The azimuth time t relative to the scene start t_0 is now also available, enabling calculation of the azimuth index:

$$I_a(t_D) = \frac{(t_D - t_0)}{\delta_t} \qquad \textbf{(22)}$$

where

$I_a(t_D)$ = azimuth image index at time t_D

t_D = azimuth time satisfying Doppler condition

t_0 = azimuth start time ($AzTimeFirst_{ML}$)

δ_t = azimuth sample interval in time ($LineTimeInterval_{ML}$)

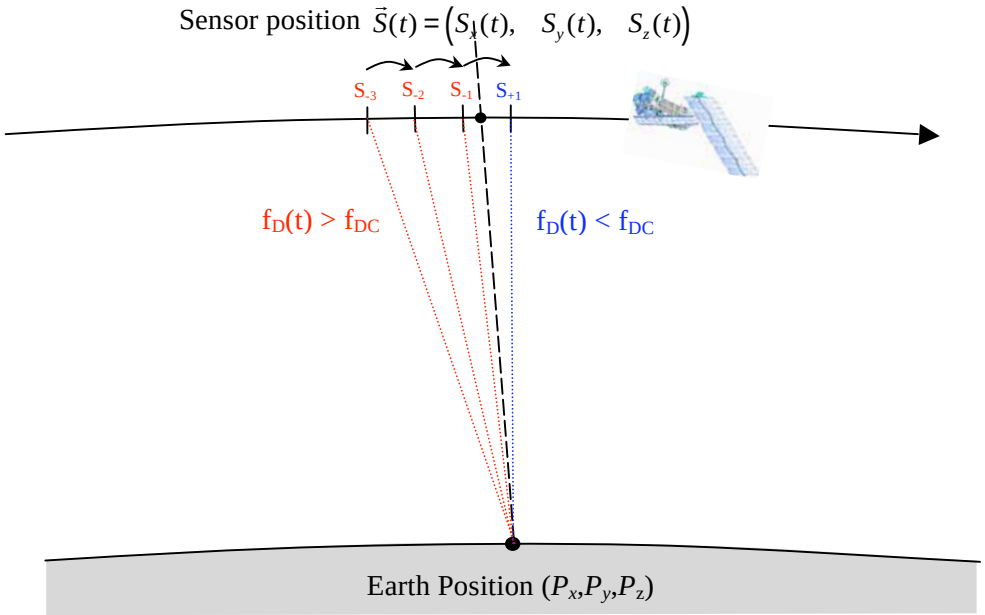


Figure 3: Determining the sensor position (and time) when a given point was imaged

4.6.2.2 Slant Range Index Computation

The slant range $R(t_D)$ between $S(t_D)$ and P at time t_D is calculated as follows:

$$R(t) = \left|\vec{P} - \overrightarrow{S(t_D)}\right| = \sqrt{(P_x - S_x(t_D))^2 + (P_y - S_y(t_D))^2 + (P_z - S_z(t_D))^2} \quad \textbf{(23)}$$

For slant range images (IMS, APS, or a WSS mosaic), the range index for the input image at time t_D is calculated as:

$$I_r(P) = \frac{(R(t_D) - r_0)}{\delta_r} \quad \textbf{(24)}$$

where

$I_r(P)$ = range image index for point P

$R(t_D)$ = slant range satisfying Doppler condition

r_0 = near range ($FastTimeNear_{ML}$ converted from time to distance)

δ_r = range pixel spacing ($RgSpacing_{ML}$)

4.6.3 Localisation Improvement

4.6.3.1 "Bistatic" Azimuth Bias Correction

Inspection of early ENVISAT ASAR imagery showed that a sizable but variable azimuth geolocation bias was present in all PF-ASAR products. The magnitude of the bias is generally approximately 20m, and can be easily corrected within the geolocation process [14].

It was determined that this bias was due to inconsistencies in interpretation of azimuth timing annotation conventions. SAR processors often advertise their level 1 output product as annotated with a Zero-Doppler time convention - geolocation algorithms then set the reference Doppler to zero to determine the azimuth coordinate of a given location on the Earth. The SAR processing in some cases however does not truly annotate in a Zero-Doppler convention, as the distance that the satellite moved between pulse transmission and reception is in some processors not considered in the calculation.

During SAR image focusing, the PF-ASAR processor transforms the image matrix to "Zero-Doppler", shifting each echo receive time to Zero-Doppler time. However, the time interval between ASAR pulse transmission and echo reception is not compensated. That interval is not Doppler-dependent - its extent depends on the swath imaged (affecting the slant range "fast" time), and can be understood as a "leakage" of range fast time into azimuth slow time. The issue is strictly an annotation convention, and can be easily compensated in post-processing during geolocation. Considering an ideal zero-Doppler case, the azimuth "bistatic" effect is illustrated in Figure 4. The relationship between slant range distance r_i and slant range "fast" time t_i is:

$$t_i^{range} = \frac{2 \cdot r_i}{c} \quad \textbf{(25)}$$

where c is the speed of light. The slant range "fast time" corresponds to the time interval between pulse transmission and echo reception. To translate between time annotation conventions, the receive time is retrieved from Zero-Doppler time by adding half of the slant range "fast time":

$$t_i^{receive} = t_i^{Zero-Doppler} + \frac{1}{2} \cdot t_i^{range} \quad \textbf{(26)}$$

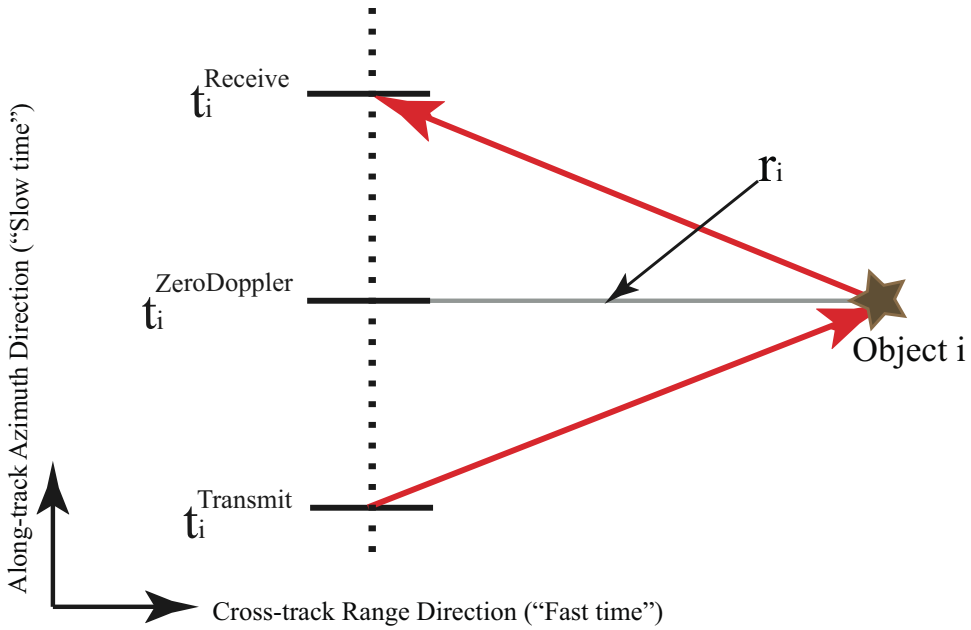


Figure 4: Azimuth "Bistatic" Effect in idealised Zero-Doppler Case

Once that correction is applied during geocoding, the location-dependent azimuth time shift may be calculated and compensated. Geolocation and geocoding proceeds otherwise normally. Comparisons of geolocation accuracies achieved with and without this compensation are presented in [15]. Note that the bias is larger at longer ranges (e.g. beam IS7) than shorter (e.g. beam IS1), also for different ranges within the swath of a single product. For that reason, the correction cannot be applied *en bloc* as a rigid shift. Due to its inherent geometry-dependency, it must instead be applied during geolocation.

Concerning the azimuth Doppler time solution t_D, we add the correction factor to produce an azimuth time t_{Dc} that has been corrected for the "bistatic" effect:

$$t_{Dc} = t_D + \frac{1}{2} \cdot t_i^{range}$$

The corrected azimuth raster index value is then:

$$I_a(t_{Dc}) = \frac{(t_{Dc} - t_0)}{\delta_t} \qquad \textbf{(27)}$$

where

$I_a(t_{Dc})$ = azimuth image index at time t_{Dc}

t_{Dc} = azimuth time satisfying Doppler condition *corrected* for bistatic bias

t_0 = azimuth start time ($AzTimeFirst_{ML}$)

δ_t = azimuth sample interval in time ($LineTimeInterval_{ML}$)

After correction for the "bistatic" effect, for slant range images (IMS, APS, or a WSS mosaic), the range index for the input image at the *corrected* azimuth time t_{Dc} is calculated from $R(t_{Dc})$ as:

$$I_r(P) = \frac{(R(t_{Dc}) - r_0)}{\delta_r} \qquad \textbf{(28)}$$

where

$I_r(P)$ = range image index for point P

$R(t_{Dc})$ = slant range at corrected azimuth time t_{Dc}

r_0 = near range ($FastTimeNear_{ML}$ converted from time to distance)

δ_r = range pixel spacing ($RgSpacing_{ML}$)

4.6.3.2 Intermittent Range Bias

Experience has shown that a (presently unresolved) intermittent range bias exhibits itself in a minority of PF-ASAR products [13]. To date, only products produced through the Range-Doppler processing chain have been found to be subject to this error. No products produced using the SPECAN input and processing chain have yet been shown to be susceptible to the problem. The product types produced by the respective input/processing chains are listed in Table 11.

Table 11: Input and Processing Chains for PF-ASAR Products

Input & Processing Chain	PF-ASAR Product
Range-Doppler	IMS, IMP, APS, WSS
SPECAN	IMM, APP, APM, WSM

Strategies to avoid or mitigate such errors (beyond choice of product type) may be provided in an updated version of this document, as they become available.

4.6.4 Ground Range Index Computation (when necessary)

In the case of ground range imagery with multiple slant/ground range polynomials references (IMM/APM/WSM), the neighbouring azimuth references are first determined, yielding two slant/ground range polynomials.

The ground to slant reference found to be valid *previous* to the current azimuth line is called $GTS_P(t_P)$ (P for previous) where $t_P \leq t_j$.

The ground to slant reference found to be valid *after* the current azimuth line is called $GTS_N(t_N)$ (N for next) where $t_N > t_j$.

The ground range IMP and APP products have only a single reference polynomial, so this step can be skipped: one can use the single ground to slant reference available (here called GTS_P) directly.

The known slant range position r is transformed using the references into a ground range solution as:

$$g_P = GTS_P^{-1}(r) \quad \textbf{(29)}$$

and similarly also using the second reference as:

$$g_N = GTS_N^{-1}(r) \quad \textbf{(30)}$$

The ground range value valid at time t_j is then calculated using linear interpolation as:

$$g = g_P + \frac{g_N - g_P}{t_N - t_P} \cdot \left(t_j - t_P\right) \quad \textbf{(31)}$$

For ground range images (IMP, IMM, APP, APM, WSM, or GM1), the range index for the input image at time *t* is calculated as:

$$I_r(P) = \frac{(g - GR_0)}{\delta_g} \quad \textbf{(32)}$$

where

$I_r(P)$ = range image index for point *P*

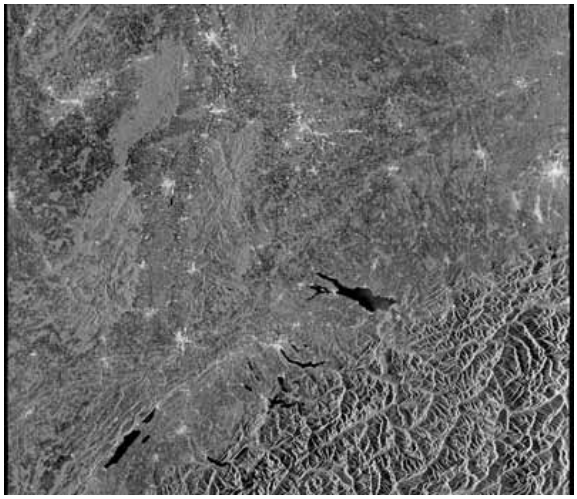
g = ground range solution corresponding to point *P*

GR_0 = ground range reference

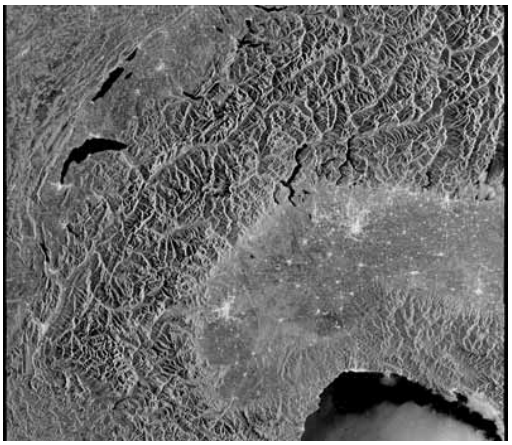
δ_g = ground range sampling interval

As a test of the above (or any) algorithm to deal with the sliding definition of ground range with multiple updates along the azimuth dimension, it can be useful to terrain-geocode two azimuth adjacent PF-ASAR products and overlay the results. Given that both products are generated from the same image acquisition, if any shifts are noted between the two terrain-geocoded GTC products, they are caused by systematic influences within the SAR focussing & geocoding processing chain. No relative shift indicates that the software is performing correctly. The test is therefore recommended before certifying a processor to be capable of accurately geocoding IMM, APM, WSM, or GM1 products.

An example of such an overlay for an ASAR image acquisition over Switzerland is shown in Figure 5.



(a) Northern WSM Product



(b) Southern WSM Product

(c) Azimuth-adjacent WSM product GTC Overlay: northern product contributes magenta=red+blue, while southern product contributes green; overlaid region appears R+G+B=grey

Figure 5: Overlay of terrain-geocoded azimuth-adjacent medium resolution PF-ASAR products as test of systematic geocoding fidelity

4.6.5 Resampling

The above sections describe how the azimuth and range indices of the input radar geometry image are derived via geolocation for a point on the Earth's surface P. Now that those indices are known, that point in the image may be resampled from radar geometry into the map geometry in question. An appropriate resampling method, such as nearest-neighbour, bilinear, or clipped cubic convolution may be employed for this purpose.

Nearest neighbour resampling proceeds by rounding the range and azimuth coordinate values to the nearest integer value, and transferring the radar image content from that location:

$$NI_r = (\text{int})\ (I_r + 0.5)$$

$$NI_a = (\text{int})\ (I_a + 0.5)$$

$$\sigma_{E,N} = \sigma\,(NI_r\ ,\ NI_a) \qquad \textbf{(33)}$$

In bilinear resampling, the four neighbouring values surrounding the (fractional) coordinate $(I_r\ ,\ I_a)$ contribute to the estimate made for that location. The neighbouring grid points are identified:

$$I_r^{\,0} = (\text{int})\ (I_r)$$

$$I_r^{\,1} = I_r^{\,0} + 1$$

$$I_a^{\,0} = (\text{int})\ (I_a)$$

$$I_a^{\,1} = I_a^{\,0} + 1$$

Linear weights scaled from zero to one encapsulate the distance to each neighbouring "corner":

$$W_r = I_r - I_r^{\,0}$$

$$W_a = I_a - I_a^{\,0}$$

$$CW_r = 1 - W_r$$

$$CW_a = 1 - W_a$$

By applying the weights, the bilinear resampling image content estimate is made:

$$\sigma_{E,N} = CW_r \bullet CW_a \bullet \sigma(I_r^{\,0}, I_a^{\,0}) + W_r \bullet CW_a \bullet \sigma(I_r^{\,1}, I_a^{\,0}) + CW_r \bullet W_a \bullet \sigma(I_r^{\,0}, I_a^{\,1}) + W_r \bullet W_a \bullet \sigma(I_r^{\,1}, I_a^{\,1}) \qquad \textbf{(34)}$$

Once the image has been resampled for all desired points and written as output, the geocoding process has completed its task.

4.6.6 Calculation of Absolute Location Error (ALE)

Points within an image that can be readily identified can have their positions both *predicted* via geometry, producing image index coordinates $(I_r\ ,\ I_a)$, as well as measured directly from the image $(M_r\ ,\ M_a)$.

The absolute location error is defined as the difference between the retrieved measured and predicted coordinates. The range error Δr_s (in image product's range samples) is:

$$\Delta r_s = I_r - M_r \qquad \textbf{(35)}$$

Note that this error is expressed in ground range samples for ground range products, and slant range samples for slant range products.

Similarly in the azimuth dimension, the error Δa_s (in image product azimuth samples) is:

$$\Delta a_s = I_a - M_a \quad \textbf{(36)}$$

The errors can be expressed as distance in metres rather than samples, as:

$$\Delta r_d = \Delta r_s \bullet RgSpacing \quad \textbf{(37)}$$

$$\Delta a_d = \Delta a_s \bullet AzSpacing \quad \textbf{(38)}$$

The absolute location error (ALE) incorporates the error in both dimensions as:

$$ALE = \sqrt{\Delta r_d^2 + \Delta a_d^2} \quad \textbf{(39)}$$

The ASAR instrument and its ground processing system have succeeded in providing products with unprecedented geometric accuracy for a civilian spaceborne SAR system. The best tests of systemic geometrical accuracy use the IMS product type, as this is closest to the radar's native geometry, and maximises the azimuth resolution, enabling the retrieval of M_a at the highest fidelity. Note however, that no significant systematic biases have been found between, for example, the IMS, IMP, and IMM product types [13]. IMG products were also found to be consistent (at subsample level) with IMS/IMP/IMM products when the radar-geometry products were ellipsoid-geocoded into the map projection annotated in the IMG product. The same good correspondence was found between APS, APP, APM, and APG products.

Many tests have shown that the ALE is generally within two product samples – exceptional cases are caused by an intermittent range bias that is still under study (see Section 4.6.3.2).

4.6.7 Impact of Local Height Variations and Ellipsoid- vs. Terrain-Geocoding

The effect of height errors within a DEM or even complete ignorance of local height values (e.g. as is the case in ellipsoid-geocoded "GEC" products) on geolocation accuracy is a horizontal shift whose magnitude depends on the incidence angle of the location under study. Vertical errors in hypsometry (height measurement) induce horizontal errors in planimetry (location information) of the product output from the geocoder.

DEM height *over*estimations cause the radar slant range estimate to be lower than the correct value: the geocoder therefore incorrectly retrieves raster content from a location closer to the image's *near* range edge than appropriate. DEM height *under*estimations cause the radar slant range estimate to be higher than the correct value: the geocoder therefore incorrectly retrieves raster information from a location closer to the *far* range edge than appropriate. Those planimetric shifts can apply at a local level within a DEM or at a scene level (e.g. for a flat image of the Netherlands that is ellipsoid-geocoded using an incorrect mean height value).

4.7 Output Annotations

Before geocoding begins, the parameters listed in Table 4 must be extracted from the ASAR product and deposited into data structures, permitting them to be accessed during initialisation and processing.

After geocoding is completed, it is useful for users to have parameters describing the geometry annotated with the output geocoded product.

Important output geometry parameters are listed Table 12. The corresponding names used in this document are shown, where applicable.

Table 12: Important output geometry parameters

Name Used in this Document	Description		Units
$Width_{GTC}$	Number of output sample columns		[samples]
$Height_{GTC}$	Number of output sample rows		[samples]
$RgSpacing_{ML}$	Range sample spacing		[m]
$AzSpacing_{ML}$	Azimuth sample spacing		[m]
$FastTimeNear_{ML}$	2-way slant range time		[seconds]
$AzTimeFirst_{ML}$	Azimuth start time		[mjd]
$AzTimeLast_{ML}$	Azimuth stop time		[mjd]
	Easting at reference sample		[m]
	Northing at reference sample		[m]
	Spacing in Easting		[m]
	Spacing in Northing		[m]
	Map geometry	Projection Type	string enumerated type, e.g. one of GEO/CHOM/TM/PS/LCC
		False Easting	[m]
		False Northing	[m]
		Scale Factor	[-]
		Central Meridian(s)	[°]
		Standard Parallel(s)	[°]
	Ellipsoid	Semi-major axis	[m]
		Semi-minor axis	[m]
	Datum Shift	Translation (x,y,z)	[m]
		Rotation (x,y,z)	[°]
		Scale Parameter	[-]

5 REFERENCES

[1] ESA SPPA Manager, ENVISAT *ASAR AP Product Quality Disclaimer*, ENVI-GSOP-EOGD-QD-05-0082, http://envisat.esa.int/dataproducts/availability/disclaimers/PQD_0082ASA_all.pdf

[2] ESA CFI Team, *Earth Explorer Mission CFI Software Release Notes*, Issue 5.6, Feb. 14, 2007.

[3] Barstow R. et al., *ENVISAT-1 Products Specifications; Volume 8: ASAR Products Specifications*, Ref. PO-RS-MDA-GS-2009, Issue 4, Rev. B, May 8, 2007.

[4] Curlander J. and McDonough R., Synthetic Aperture Radar: Systems & Signal Processing, John Wiley & Sons, New York, 1991.

[5] Dyer J., *Generalised Multistep Methods in Satellite Orbit Computation*, JACM, Volume 15, Issue 4, Oct. 1968, pp. 712-719.

[6] Escobal P., Methods of Orbit Determination, John Wiley & Sons, New York, 1965.

[7] Meier E., Frei U., Nüesch D., *Precise Terrain Corrected Geocoded Images*, chapter in SAR Geocoding: Data and Systems, ed. G. Schreier, Herbert Wichmann Verlag GmbH, 1993.

[8] McLeod I. et al., *ENVISAT-1 Products Specifications; Volume 5: Product Structures*, Ref. PO-RS-MDA-GS-2009, Issue 3, Rev. C, October 16, 1997.

[9] Press W., Teukolsky S., Vetterling W., Flannery B., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1992.

[10] Rosich B., Meadows P., *Absolute Calibration of ASAR Level 1 Products Generated with PF-ASAR*, ENVI-CLVL-EOPG-TN-03-0010, ESA-ESRIN, Issue 1, Revision 5, Oct. 7, 2004.

[11] Schubert A., Small D., *ASAR WSS Mosaic Algorithm Description*, ESA Report Reference RSL-WSS-MOSAIC-AD, Issue 1.1, Jan. 22, 2008.

[12] Schubert A., Small D., Rosich B., Meier E., *ASAR WSS Product Verification Using Derived Image Mosaics*, Proc. Envisat Symposium 2007, Montreux, Switzerland, April 23-27, 2007 (ESA SP-636, July 2007).

[13] Small D., Schubert A., Rosich B., Meier E. *Geometric and Radiometric Correction of ESA SAR Products*, Proc. Envisat Symposium 2007, Montreux, Switzerland, Apr. 23-27, 2007 (ESA SP-636, July 2007). 6p.

[14] Small D., Rosich B., Schubert A., Meier E., Nüesch D., *Geometric Validation of Low and High-Resolution ASAR Imagery*, *Proc. 2004 ENVISAT & ERS Symposium*, Salzburg, Austria, Sept. 6-10, 2004 (ESA SP-572, April 2005). 9p.

[15] Small D., Rosich B., Meier E., Nüesch D., *Geometric Calibration and Validation of ASAR Imagery*, Proc. CEOS SAR Workshop, Ulm, Germany, May 27-28, 2004.

[16] Small D., Schubert A., Krüttli U., Meier E., Nüesch D., *Preliminary Validation of ASAR Geometric Accuracy*, Proc. ENVISAT Validation Workshop, Frascati, Italy, Dec. 9-13, 2002 (ESA SP-531, August 2003).

[17] Unser M., Aldroubi A., Eden M., *B-Spline Signal Processing: Part I – Theory*, IEEE Trans. Signal Processing, Vol. 41, No. 2, pp. 821-832, Feb. 1993.