SMOS L1 Processor L1c Data Processing Model

Code :	SO-DS-DME-L1OP-0009
Issue :	2.17
Date :	28/11/17

	Name	Function	Signature
Prepared by	A. Gutierrez	Project Engineer	
	R. Castro	Project Engineer	
	P. Vieira	Project Engineer	
	G. Lopes	Project Engineer	
	J. Barbosa	Project Engineer	
Checked by	A. Gutierrez	Quality A. Manager	
Approved by	J.Barbosa	Project Manager	

DEIMOS Engenharia
Av. D. João II, Lote 1.17, Torre Zen, 10°,
1998-023 Lisboa, PORTUGAL
Tel: +351 21 893 3013
Fax: +351 21 896 9099
E-mail: mailto:deimos@deimos.com.pt

This page intentionally left blank

Document Information

Contract Data	Classification
Contract Number: 4000101241/10/I-AM	**Internal** ☑ **Public** ☐ **Industry** ☐ **Confidential** ☐
Contract Issuer: ESA ESRIN	

Internal Distribution		
Name	Unit	Copies

External Distribution		
Name	Organisation	Copies
Steven Delwart	ESA	1
Raffaele Crapolicchio	ESA	1

Archiving	
Word Processor:	MS Word 2000
File Name:	SO-DS-DME-L1OP-0009_ DPML1c-Data-Processing-Models_E2_R17.docx
Archive Code:	SO-DS-DME-L1OP-0009

Document Status Log

Issue	Change description	Date	Approved
	L1PP Document		
0.1	Table of Contents	2005-06-01	
1.0	First Version for CDR	2005-06-30	
1.1	Update after CDR RIDs	2005-08-31	
1.2	Update after L1PP implementation	2006-04-07	
1.3	Final update for Phase 2 completion	2006-06-07	
2.0	Updates for L1PP V2R	2006-11-17	
2.1	Updates for L1PP V3.5	2007-07-15	
2.2	Updates for L1PP V4: - Details on browse generation - Details on TEC interpolation - Defined baseline for IRI and IGRF models - Defined Sun Tails flagging strategy	2007-11-16	
2.3	Updates for L1PP V5R: - Corrected details on L1c Browse generation - Included usage of Mispointing ADF in attitude initialisation	2008-03-31	
2.4	Updates for L1PP V6R: - Added details on Analytical Footprint computation in section 3.2.5 and 5.11 - Included extra computation in Attitude initialisation to derive rotation matrix from BFP to EF	2008-07-31	
2.5	Final update for Phase 4 completion (L1PP V2.0): - Corrected Blackman factor in pseudo-code in section 5.7.4 - Updated Geometric Rotation formulation in sections 3.1.4 and 5.10 - Clarified inputs for fast method of Footprint computation in section 5.11 -Clarified Gibbs 2 procedure in section 3.2.2 and 5.8.4	2008-12-12	
2.6	Update for L1PP V2.2: - Added Sunglint area flag computation function in section 5.12	2009-07-24	

Issue	Change description	Date	Approved
2.7	Update for L1PP V3.4: - Updated Faraday rotation angle equation according to new version of [RD.14]. Sections 3.1.3 and 5.10.4 - Changed integration time applicable in theoretical radiometric accuracy according to KP3 experimental results. Sections 3.2.3 and 5.9.4	2010-05-31	
2.7b	Update for the beginning of the SMOS operations phase: - Introduced clarification on theoretical radiometric accuracy computation for pure and mixed scenes in section 5.9.4 - Introduced clarification on BFP and MISP rotation angles computation by using all rotations contained inside each file in section 5.2.4	2010-07-15	
2.8	Update for L1PP V3.5: - Added RFI Flagging routines	2010-10-29	
2.9	Update for L1PP V5.0 - Updated border flag computation with the distance to the Unit Circle aliases (“suspenders and belt”) in section 5.12 - Changed the sun circle radius flagging value to be configurable and added explanation on RFI flags in section 5.12	2011-05-06	
2.10	Update for L1PP V5.5: - Added Open Issue regarding apodisation windows used in L1	2011-11-29	
2.11	Update for L1PP V6.0 - Updated RFI Flagging conditions to inherit RFI snapshot flag from L1b/L1a products instead of computing it in L1c	2012-11-29	
L1OP Document			
2.12	Update for L1OP v6.0.0 - Removed all references to L1PP - Section 5.7.4: Removed Strip Adaptive Method tocompute the apodisation factor - Section 7. Annex 1: Global Data Types, CONSTANTS AND FUNCTIONS has been removed.	2013-05-02	
2.13	Update for L1OP v6.1.0: Updated height used for Geomagnetic Model from 450km to 400km	2013-09-10	

Issue	Change description	Date	Approved
2.14	Update for L1OP v6.2.0: Updated with new RFI flagging mechanism (Section 5.12.4) Updated obsolete references to baseline definition (now in TGRD document)	2014-03-25	
2.15	Update for L1OP v7.1.0 Updated with description of Gibbs 2 scene addition (Sections 3.2.2 and 5.8.4) Updated with new RFI contamination flags description (Sections 3.4 and 5.12.4)	2015-02-12	
2.16	Updated with comments from L1OP v711 FAT New System Concept and Geolocation Modules diagrams Better explanation of RFI tails flagging in Section 4.14 Update of the IGRF model version to the 12th generation	2017-01-31	
2.17	Update for L1OP v7.2.0 Updated for optimized RFI contamination level flags computation (Section 5.1.2)	2017-11-28	

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1. Purpose

The main purpose of this document is to provide a detailed definition of the processes and algorithms contained in the L1b to L1c processing module.

The document begins with a general overview of algorithms to be used in the Geolocation, Ionospheric Correction and Apodisation Window Computation modules, its main tasks and its role in the Level 1 Processor. Following the identification of a high-level architecture, a further top-down decomposition is presented to cover the whole set of algorithms involved in the L1C data processing which, in turn, will be described in detail.

1.2. Scope

The scope of this document is to describe the algorithms involved in the processing of the Geolocation, Ionospheric Correction and Apodisation Window Computation modules, providing both architectural and functional definitions of the main features identified.

1.3. Acronyms and Abbreviations

For the list of acronyms, please refer to the "Directory of Acronyms and abbreviations" [AD.4].

1.4. Applicable and Reference Documents

1.4.1. Applicable Documents

Ref.	Code	Title	Issue & Date
AD.1	Removed		
AD.2	SO-RS-ESA-PLM-0003	SMOS System Requirements Document	***3.0***
AD.3	Removed		
AD.4	SO-LI-CASA-PLM-0094	Directory of Acronyms and abbreviations	***3.0*** **12/22/03**
AD.5	ECSS-E-40B	ECSS E-40 Software Engineering Standards	
~~AD.6~~	~~SO-DS-DME-L1OP-0011~~	~~SMOS L1 Processor Algorithm Theoretical Baseline Definition~~	~~***2.10***~~ ~~*29/10/10*~~
AD.7	SO-DS-DME-L1OP-0007	SMOS L1 Processor L0 to L1a Data Processing Model	***2.22*** *28/11/17*

AD.8	SO-DS-DME-L1OP-0008	SMOS L1 Processor L1a to L1b Data Processing Model	***2.23*** *28/11/17*
AD.9	SO-TN-IDR-GS-0005	SMOS Level 1 and Auxiliary Data Products Specifications	***6.4*** *18/10/17* *(Preliminary Version)*

Table 1 – Applicable Documents

1.4.2. Reference Documents

Ref.	Code/Author	Title	Issue & Date
RD.1	EE-MA-DMS-GS-0001	Earth Explorer Mission CFI Software MISSION CONVENTIONS DOCUMENT	***3.7.3*** *07/05/10*
RD.2	PE-TN-ESA-GS-0001	Earth Explorer Ground Segment File Format Standard	***1.4*** *13/06/03*
RD.3	EE-MA-DMS-GS-0002	Earth Explorer Mission CFI Software GENERAL SOFTWARE USER MANUAL	***3.7.3*** *07/05/10*
RD.4	Removed		
RD.5	Removed		
RD.6	Removed		
RD.7	David M. Le Vine & Saji Abraham	*"Faraday rotation and passive microwave remote sensing of soil moisture from space"*	2000
~~RD.8~~	~~SO-DS-DME-L1PP-0006~~	~~SMOS L1 System Concept~~	~~***2.9*** *29/10/10~~
RD.9	SO-TN-DME-L1PP-0024	SMOS L1 Full Polarisation Data Processing	***1.6*** *16/07/07*
~~RD.10~~	~~SMOS-DMS-TN-5100~~	~~Adaptive Apodisation Function Development Technical Note~~	~~***1.2***~~
RD.11	P. Waldteufel, G. Caudal	*"About Off-Axis Radiometric Polarimetric Measurements"*, IGARSS	2002
RD.12	INETI-SMOS-1-2004-TN-5100-002	*"On the use of 2D KAISER Functions on the SMOS/MIRAS adaptive striping problem"*	2004

Ref.	Code/Author	Title	Issue & Date
RD.13	SO-TN-DME-L1PP-0172	*Analytical Pixel Footprint Technical Note*	***1.0*** *18/04/08*
RD.14	M. Zundo, B.Duesmann SO-TN-ESA-GS-5873	*On-ground BT Frame of Reference TN*	***3.3*** *24/05/10*
RD.15	SO-TN-DME-L1PP-0241	RFI Algorithms in L1PP	***1.2*** *26/10/11*
RD.16	SM-TN-AURO-L1OP-0001	TN on the L1OP RFI Flags	***2.1*** *27/05/17*
RD.17	SO-TGRD-DME-L1OP-0023	SMOS L1 Processor Table Generation Requirements Document	***3.11*** *28/11/17*

Table 2 – Reference Documents

Ref.	Article
RA.01	*Sun Effects in 2-D Aperture Synthesis Radiometry Imaging and Their Cancellation* (A. Camps, M. Vall-llossera, N. Duffo, M. Zapata, I. Corbella, F. Torres, and V. Barrena), IEEE Transactions on Geoscience and Remote Sensing, 42 (6): 1161-1167. ISSN: 0196-2892, 2004
RA.02	*Direct Least Squares Fitting of Ellipses* (Fitzgibbon, A.W. and Pilu, M. and Fisher, R.~B.), IEEE Transactions on Pattern Analysis and Machine Intelligence, 21 (5): 476-480. May 1999

Table 3: Reference articles

2. L1B TO L1C IN THE SMOS L1 PROCESSOR

The last step of the processing chain shall perform a combination of the Brightness Temperature snapshot values retrieved previously into swath format. Instead of being ordered by integration time and containing the info for each pixel on the FOV, they shall be ordered by pixels identified in a Fixed Earth Grid, each of them containing the Brightness Temperature values related to different incidence angles and different integration times. This step shall also perform the required apodisation of the BT values.

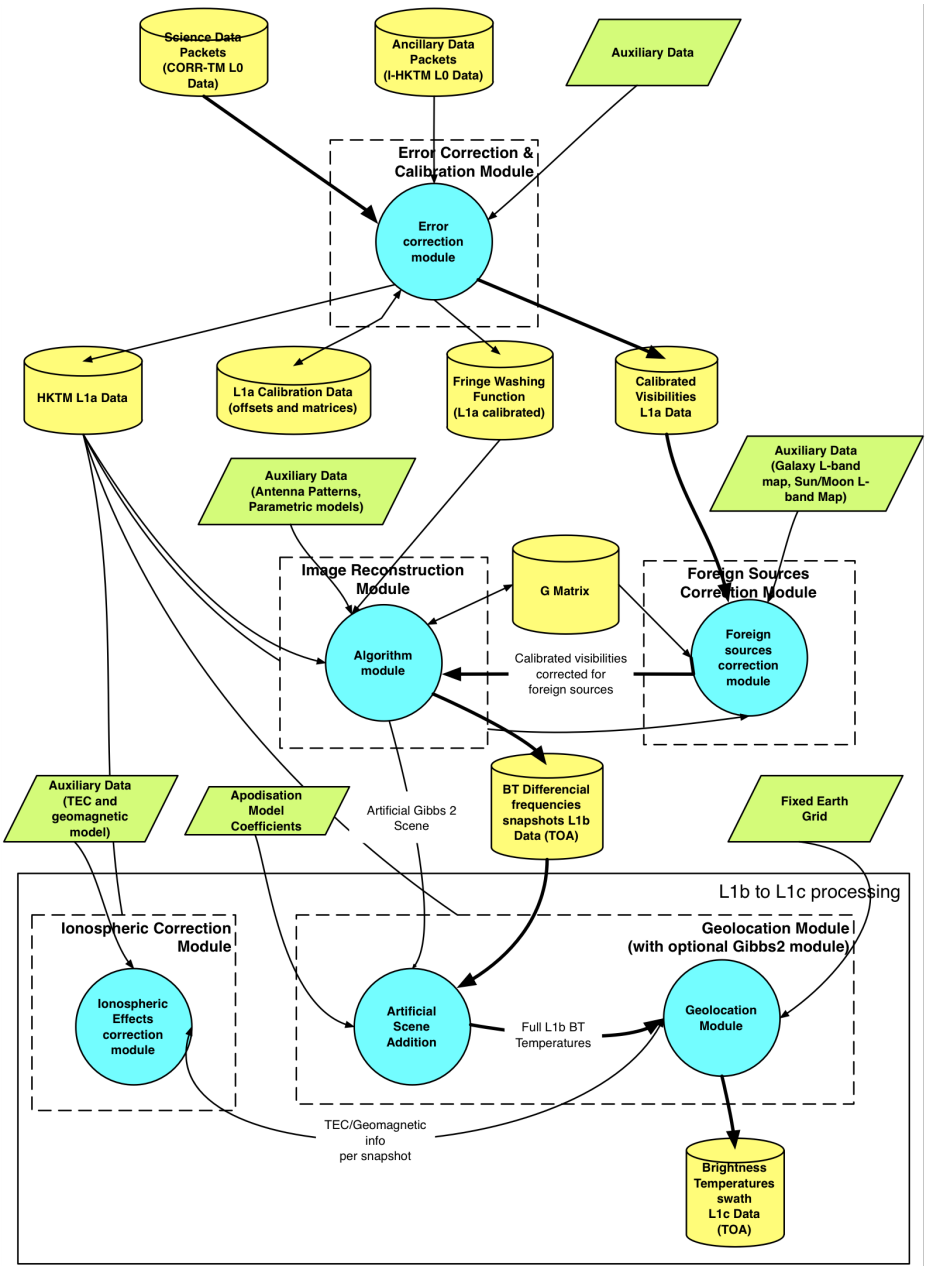

Figure 1: L1 Processor Complete Dataflow

2.1. Geolocation module

The Geolocation module shall compute the BT values and their incidence angles in the direction of the required Earth fixed points, while at the same time computing the characteristics of the footprint (area and orientation). This is achieved by applying a Discrete Fourier Transform on the L1b BT frequency components and particularising over the Fixed Earth grid points. Using "normal" processing, the pixel size for a same geographical location changes in size in consecutive snapshots, as the apodisation applied in the DFT is the same regardless of the point position. This effect shall make it necessary to provide footprint information to the L2 users so that it can be taken into account.

The DFT allows the apodisation and interpolation to the Earth Fixed grid in the same processing step.

This module shall compute sets of temperature values, H and V for dual-pol operation (real valued), and H, V and HV for full-pol operation (HV is complex valued). The BT values are produced in consecutive integration times. The Ionospheric Correction module will compute the geomagnetic angles necessary for the rotation correction from instrument to ground and store them as part of the L1c product.

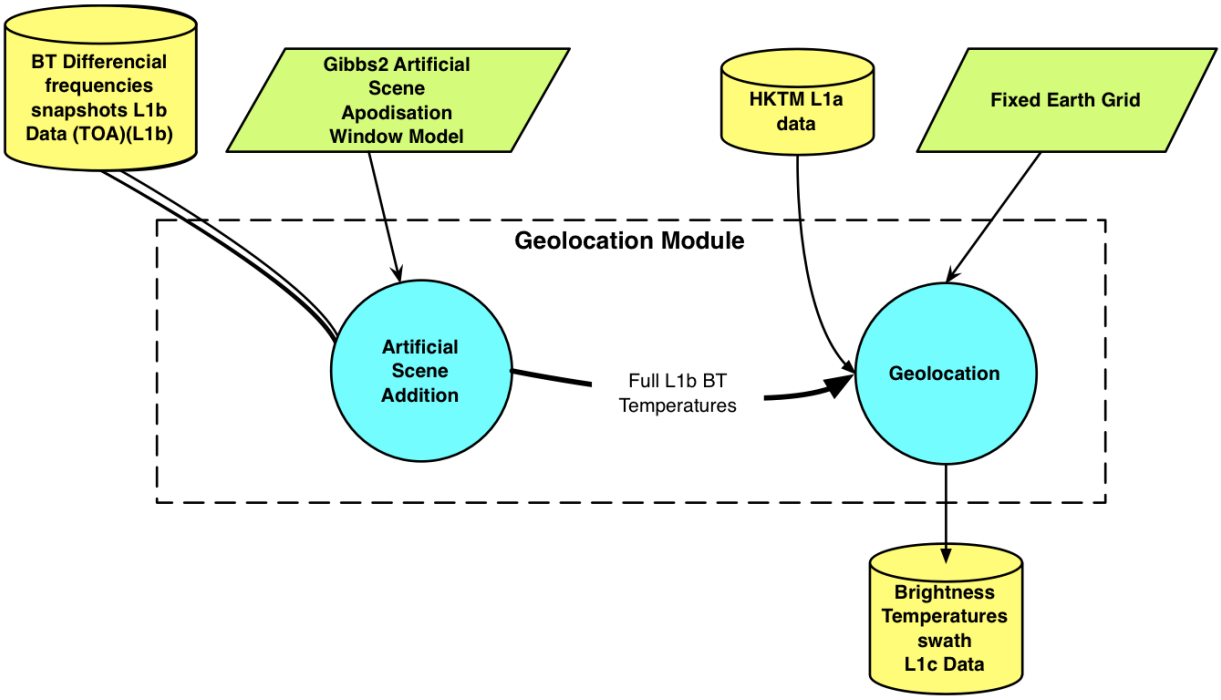

Figure 2: Geolocation Module

In Gibbs 1 mode, the Artificial Scene Addition only adds the average Earth temperature (a single scalar) to the zero baseline. In Gibbs 2 mode, the artificial Gibbs 2 scene temperatures are apodized in the same way as the measurements (Blackmann Window) and are added after the Discrete Fourier Transform is applied to the apodized data.

2.2. Ionospheric Correction Module

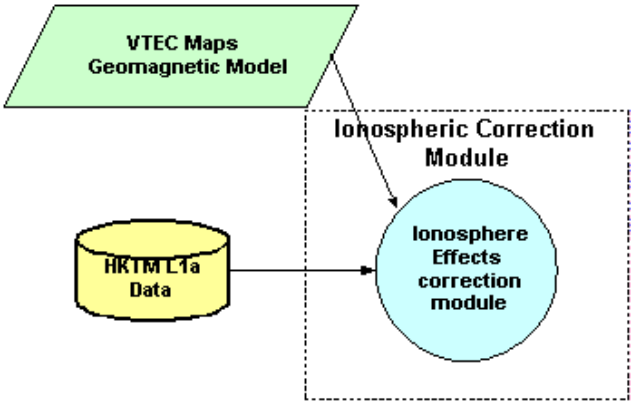

Figure 3: Ionospheric Correction Module

This module shall compute one TEC value and one geomagnetic vector value per each snapshot, so that the subsequent (L2) Faraday correction for each footprint can be computed using these values. It is considered that TEC and geomagnetic variations are sufficiently small within the scene FOV such that one mean value of each can be considered as the baseline. The rotation due to the change of reference from instrument to ground shall also be computed here on a per-pixel basis.

The information required for the correction shall be extracted from specific auxiliary data combined with HKTM L1a data. It shall be attached internally to the L1c data, in the initial time-ordered list of snapshots parameters.

3. DESCRIPTION OF ALGORITHMS

3.1. Ionospheric Correction module

This module shall compute a set of values on a snapshot basis, which are required for computing the Faraday rotation angles for each pixel within that snapshot. The values are the Total Electron Content (TEC) and the geomagnetic angles at boresight, which shall be passed to the Geolocation module for computation of the Faraday rotation angle.

This Faraday rotation angle accounts for the polarisation mixing that the Brightness Temperatures suffer when passing through the Ionosphere on the path from each pixel to the MIRAS instrument.

The computation of those values shall be performed based on existing models, and as an additional possibility, the TEC can be obtained from a fast "IGS Combined" product available every 24h.

Ionospheric correction performs then the computation of the Total Electron Content and the geomagnetic angles. Later on, the values are particularised at each pixel by simply using the incidence (θ_g) and azimuth (ϕ_g) angles from the spacecraft to the pixel (shown in Fig. 5).

3.1.1. Computation of TEC

As mentioned before, the TEC can be obtained from several sources, the first one is the IRI2001 model available as well from NSSDC in FORTRAN code, the second one is the IGS combined TEC map produced by UPC and available daily as Auxiliary Data File at ftp://gage152.upc.es/rapid_iono_igs.

The IRI2001 model is based on the 11-year Sun solar cycle, and requires as input the geodetic longitude and latitude, the time and the altitude, as well as several configuration parameters. The IGS combined TEC map is an ASCII file with TEC values over a Mercator grid at an altitude of 450km and measured every 2 hours. The frequency with which this file is released is every 24 hours. The expected output in both methods is the TEC value for the given spatiotemporal coordinates expressed in TEC Units ($10^{+016}e^{-}/m^2$).

3.1.2. Computation of Geomagnetic Angles

The geomagnetic angles are computed using the IGRF 11th generation model, valid until 2015 and available as an Auxiliary Data File, and the IGRF FORTRAN code available from NSSDC. The required inputs at any time are the S/C geodetic longitude and latitude, the time in decimal years and the altitude. The expected outputs shall be the magnetic field strength in Tesla (F), as well as the magnetic inclination (I) and declination (D) in degrees.

3.1.3. Computation of the Faraday rotation angles

Computation of Faraday rotation (ω) for each pixel is computed using Eq.84 of [AD.6], but corrected according to [RD.14] sign convention and reference (i.e. from surface to antenna), with the retrieved pixel observation angles, TEC and geomagnetic data.

$$\omega = -6950 \times F \times TEC \times \left[\sin I + \cos I \times \tan \theta_g \times \cos\left(\phi_g - D\right) \right] \qquad \textit{Eq. 1}$$

3.1.4. Geometrical rotation

Apart from the Faraday rotation effect on the electromagnetic wave, another effect must be taken into account due to the change of reference frame between instrument and pixel. The instrument is measuring the polarised energy along two axes determined by the output ports of the LICEF receivers, and these axes do not correspond to the Horizontal and Vertical polarisation axes at pixel level. Thus it is required to compute another angle (geometrical rotation) to relate the reference change per pixel.

3.1.4.1. Waldteufel and Caudal Implementation

The geometrical rotation between the polarisation axes in the antenna frame and the pixel frame are computed as follows. The diagram below shows the angles necessary to describe a projection of point P into the antenna frame (SXYZ)

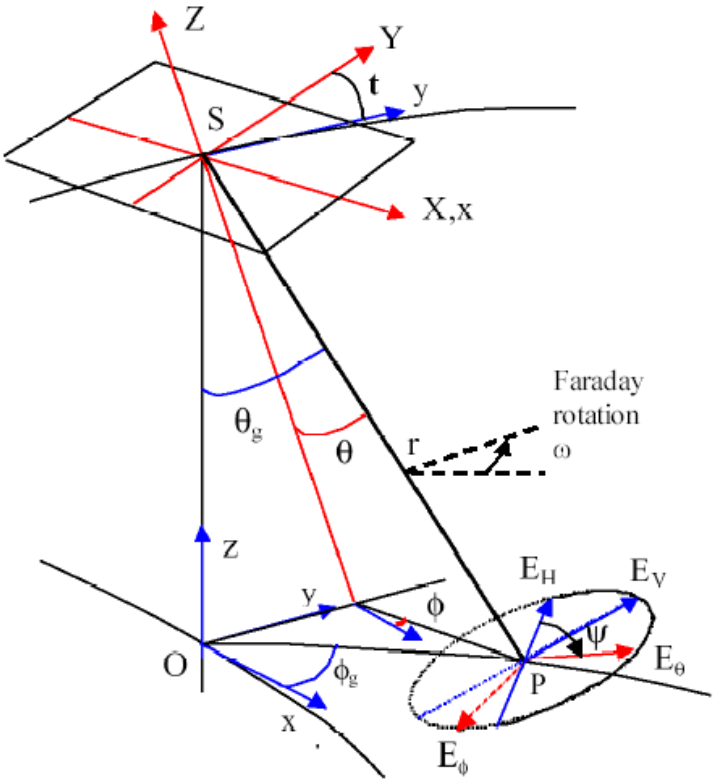

Figure 4: Geolocation and projection angles [RD.11]

Where t is the tilt angle of the instrument (32.5º), X and Y are the polarisation axes in the antenna frame, and E_H and E_V are the polarisation axes in the pixel frame. E_H is orthogonal to the PS and PO directions, and E_V is orthogonal to E_H and PS.

The angles may be computed using the following expressions, as defined in [RD.11]:

$$\theta = \arccos\left[\sin t \sin \theta_g + \cos t \cos \theta_g \right] \qquad \textit{Eq. 2}$$

$$\phi = \arcsin\left[\frac{-\sin t\cos\theta_g + \cos t\sin\theta_g \sin\phi_g}{\sin\theta}\right] \qquad \textit{Eq. 3}$$

$$\varphi = \pi - \arcsin\left[\frac{\cos t\sin\theta_g - \sin t\cos\theta_g \sin\phi_g}{\sin\theta}\right] \qquad \textit{Eq. 4}$$

Where ϕ is measured from the X axis and positive clockwise in the XYZ reference frame, and φ is the in plane geometrical rotation between polarisation frames. The above expressions are valid for the range $\phi_g \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, for the range $\phi_g \in \left[\frac{\pi}{2}, \frac{3\pi}{2}\right]$, the angles must be simply computed as:

$$\phi = -\pi - \arcsin\left[\frac{-\sin t\cos\theta_g + \cos t\sin\theta_g \sin\phi_g}{\sin\theta}\right] \qquad \textit{Eq. 5}$$

$$\varphi = \arcsin\left[\frac{\cos t\sin\theta_g - \sin t\cos\theta_g \sin\phi_g}{\sin\theta}\right] \qquad \textit{Eq. 6}$$

The final geometrical rotation value from pixel frame to antenna frame is given by $\alpha = \phi - \varphi$

The antenna frame observation angles can also be obtained using the Cartesian coordinates of the S/C and the Cartesian coordinates of the pixel, plus the instrument attitude rotation matrix (and additionally miss-pointing like Best Fit Plane deviation). Using this last method, the pixel coordinates do not need to be restricted to the Reference Ellipsoid, but they can be expanded with the local altitude value using a Data Elevation Model. This process of orthorectification shall be done with the help of the EE CFI functions.

This method is no longer used but its description is kept for historical reasons.

3.1.4.2. Duesmann and Zundo Implementation

In [RD.14] an alternative method to compute the geometric rotation angle is proposed. It consists on calculating the emission basis vectors in the Earth's surface and the Ludwig-3 polarization basis vectors on the instrument frame and from them extract the geometrical rotation angle, α. This method is the algorithm baseline.

The emission basis vectors in the topocentric frame of the pixel are calculated from the observation angles as

$$\begin{aligned} \hat{Ehe}_{x_{topocentric}} &= -\cos\phi_e \\ \hat{Ehe}_{y_{topocentric}} &= \sin\phi_e \\ \hat{Ehe}_{z_{topocentric}} &= 0 \end{aligned} \qquad \textit{Eq. 7}$$

$$\begin{aligned} \hat{Eve}_{x_{topocentric}} &= -\cos\theta_e \sin\phi_e \\ \hat{Eve}_{y_{topocentric}} &= -\cos\theta_e \cos\phi_e \\ \hat{Eve}_{z_{topocentric}} &= \sin\theta_e \end{aligned} \qquad \textit{Eq. 8}$$

To calculate the Ludwig-3 basis vectors on the satellite one must start by defining the target vector from the satellite to the pixel, $\vec{r}^{MIRAS}$,

$$\vec{r}^{MIRAS} = \begin{cases} \cos\theta_s \sin\phi_s \\ \cos\theta_s \cos\phi_s \\ -\sin\theta_s \end{cases} \qquad \textit{Eq. 9}$$

and with it define two auxiliary vectors, N_x and N_y:

$$\begin{aligned} v_{aux}^{MIRAS} &= \vec{r}^{MIRAS} - u_z^{MIRAS} \\ N_x^{MIRAS} &= \frac{v_{aux}^{MIRAS} \times u_x^{MIRAS}}{\left\| v_{aux}^{MIRAS} \times u_x^{MIRAS} \right\|} \\ N_y^{MIRAS} &= \frac{v_{aux}^{MIRAS} \times u_y^{MIRAS}}{\left\| v_{aux}^{MIRAS} \times u_y^{MIRAS} \right\|} \end{aligned} \qquad \textit{Eq. 10}$$

These auxiliary vectors are used to calculate the Ludwig-3 vectors

$$\begin{aligned} L_x^{MIRAS} &= \frac{N_x^{MIRAS} \times \vec{r}^{MIRAS}}{\left\| N_x^{MIRAS} \times \vec{r}^{MIRAS} \right\|} \\ L_y^{MIRAS} &= \frac{N_y^{MIRAS} \times \vec{r}^{MIRAS}}{\left\| N_y^{MIRAS} \times \vec{r}^{MIRAS} \right\|} \end{aligned} \qquad \textit{Eq. 11}$$

Having $\hat{Ehe}_{topocentric}$, $\hat{Eve}_{topocentric}$, L_x^{MIRAS} and L_y^{MIRAS} to the same referential, the polarization angle is obtained as shown in the following equation:

$$\alpha' = \arctan\left(\frac{\vec{Eh}_{MIRAS} \cdot \vec{L}_y^{MIRAS}}{\vec{Eh}_{MIRAS} \cdot \vec{L}_x^{MIRAS}} \right) \qquad \textit{Eq. 12}$$

For further details, please refer to [RD.14].

3.2. Geolocation module

The objective of the Geolocation process is to compute for each of the pixels observed the corresponding Brightness Temperature and observation angles, the radiometric accuracy, the Faraday and geometric rotation angles, the footprint major and minor semi-axis and a set of quality flags. The pixels shall be part of an Earth Fixed Grid (ISEA), such that the measurements are always referred to the same geodetic coordinates.

In order to compute the data, it shall use the L1b Brightness Temperature frequencies to perform a DFT in the direction of each pixel, plus obtaining a radiometric accuracy measurement of the errors induced in that measurement. It shall also compute the equivalent synthesized beam in order to extract the pixel (or footprint) spatial resolution.

3.2.1. Alias Free FOV Computation

The first step shall be to identify which pixels of the ISEA grid are contained within the extended alias-free FOV for one particular snapshot. This shall be done by first projecting the extended AF-FOV contour onto the ISEA grid and gathering the points falling inside.

After identifying which pixels of the ISEA grid are contained within the extended alias-free FOV for each snapshot, the incidence and azimuth angles in the antenna frame can be computed for each of the pixels, which also correspond to the xi-eta coordinates, through to the following equation:

$$\xi = \sin\theta\cos\phi$$
$$\eta = \sin\theta\sin\phi \qquad \textit{Eq. 13}$$

As it will be shown during the implementation, different angle conventions between the usual SMOS documentation and the CFI angle convention may show different equations. For example in the above equation θ is measured from the boresight towards the viewing direction, and ϕ is measured from the 0º axis.

The computation of the EAF-FOV is done through the selection of a geometrical boundary that follows certain requirements. These requirements can be observed in the next figure:

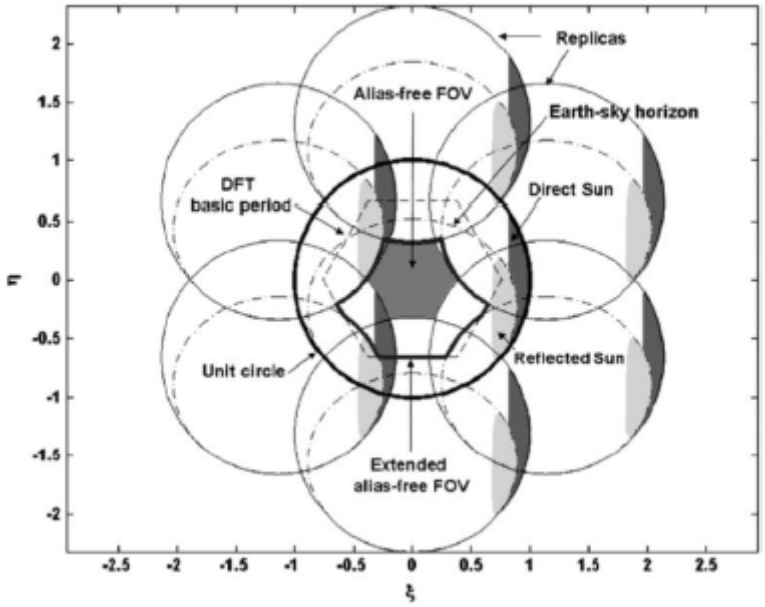

Figure 5: Areas in FOV for a 2-D aperture synthesis Y-shaped interferometric radiometer, with an antenna spacing of 0.875 wavelengths, 32º array tilt, 30º array steering and 755 km platform height [RA.01]

The exact procedure shall be detailed in section 5.

3.2.2. Pixel Brightness Temperature computation

The Brightness Temperature computation to be performed per pixel shall be a Discrete Fourier Transform, using as input the brightness temperatures frequencies $\hat{T}(u,v)$ produced as L1b output, the *(u,v)* baselines coordinates, the apodisation window coefficients *W(u,v)* and the xi-eta coordinates of that particular point:

$$T(\xi,\eta)=\frac{\sqrt{3}}{2}d^2\sum_m\sum_n\hat{T}(u_{mn},v_{mn})\cdot W(u_{mn},v_{mn})\cdot e^{j2\pi(u_{mn}\xi+v_{mn}\eta)} \qquad \textbf{Eq. 14}$$

In the nominal case, the apodisation window W coefficients shall be constant regardless of the xi-eta coordinates (e.g. Blackman window). In the case of strip adaptive, the coefficient shall be also a function of the xi-eta coordinates. The method for W computation is shown in the last chapter of this section.

In case that Gibbs correction was applied during L1b processing, this correction removed a constant flat Earth or a constant Land and a model Ocean from the reconstructed image, which must be added back after the DFT is performed. For Gibbs 1, this constant Earth or Land Brightness Temperature is reported inside each snapshot of the L1b products, and is applicable only for dual polarisation scenes, never for full pol scenes.

In Gibbs 1 mode, the Artificial Scene Addition only adds the average Earth temperature (a single scalar) to the apodized scene data. In Gibbs 2 mode, the artificial Gibbs 2 scene temperatures are apodized in the same way as the measurements (Blackmann Window) and are added after the Discrete Fourier Transform is applied to the apodized data.

3.2.3. Pixel Radiometric Accuracy computation

Additionally, the radiometric accuracy shall be computed for each pixel and polarisation. The equation for doing this is the following:

$$\Delta T^{pq}(\xi,\eta)=\frac{\Omega^{pq}\sqrt{1-\xi^2-\eta^2}}{\bar{G}^{pq}(\xi,\eta)}\frac{\sqrt{3}}{2}d^2\frac{T_{sys}^{pq}}{\sqrt{B\cdot\tau_{eff}}}\alpha_w\alpha_{ol} \qquad \textbf{Eq. 15}$$

Where the different parameters are:

- Ω^{pq} is the solid angle of the antenna, and can be approximated by $\frac{4\pi}{D}$ for the current LICEFs, where D is the averaged directivity of the LICEFs at the corresponding polarisation *pq*.
- $G^{pq}(\xi,\eta)$ is the averaged LICEF receiver directional power Gain function normalised so that it is unity at boresight. This data shall be retrieved from the antenna patterns ADF for each polarisation.
- *d* is the distance ratio between receivers (0.875)
- T_{sys}^{pq} is the averaged System Temperature measured by the PMS system, which has been used to de-normalise the L1a calibrated visibilities.

- B is the equivalent receiving frequency bandwidth in Hz, currently being 19 MHz
- τ_{eff} is the effective integration time, and is equivalent to τ / c_{eff} where τ is the integration time (1.2s for pure dual pol epochs, 0.46 for mixed dual pol epochs, and 0.64 for HV pol epochs) and c_{eff} is the coefficient that accounts for the 1-bit correlation, oversampling and hermiticity (c_{eff}=1.81).
- $\alpha_w = \sqrt{\sum_u \sum_v \frac{\left(W(u,v)\right)^2}{R(u,v)}}$ accounts for the apodisation window and redundancies in the measurements, where W is the apodisation window term for each (u,v) baseline and R is the redundancy level of that same baseline (i.e. number of times that the baseline has been measured, 1 for non-redundant baselines, greater than 1 for the rest)
- $\alpha_{ol} = \sqrt{1 + e^{-2\pi\left(\frac{f_0 - f_{01}}{B}\right)^2}}$ accounts for the local oscillator factor, where B is the bandwidth mentioned before, f_0 is the central frequency and f_{01} is the low frequency, whose values are obtained from the PLM ADF.

3.2.4. Pixel Observation Angles computation

Afterwards, the incidence (θ_p) and azimuth (ϕ_p) observation angles shall be computed. The incidence angle is the angle between the local normal at each pixel and the pixel-to-satellite direction, whereas the azimuth angle is the angle measured between the pixel-to-satellite direction projected in the local tangent plane and the local North direction. These angles shall be computed with the help of the CFI functions, using the spacecraft position and the latitude, longitude and altitude coordinates of each pixel in the Earth fixed grid.

3.2.5. Pixel Footprint Shape Computation

Beyond these computations, the pixel shape shall be obtained as the projection onto the Earth fixed Grid of the –3dB contour of the Synthetic Antenna Directional Gain.

On a first approximation, this projection shall be approximated by an ellipse, in which its major semi-axis is not necessarily oriented with the azimuth (ϕ_p) observation angle. This makes it mandatory to project at least 6 points in order to make a least squares fit of an ellipse to derive its semi-axes.

The Synthetic Antenna Directional Gain (also known as Equivalent Array Factor) may be computed in the antenna frame by following the expression:

$$AF_{eq}(\xi,\xi',\eta,\eta') = \frac{\sqrt{3}}{2} d^2 \sum_m \sum_n W(u_{mn},v_{mn}) \cdot \not{p}\left(-\frac{u_{mn}\cdot\xi + v_{mn}\cdot\eta}{f_0}\right) \cdot e^{j2\pi\left(u_{mn}(\xi-\xi')+v_{mn}(\eta-\eta')\right)} \qquad \textbf{Eq. 16}$$

Where:

- $W(u_{mn},v_{mn})$ is the apodisation function computed before
- $\not{p}$ is the fringe-washing function which accounts for the spatial decorrelation between antennas. It is calibrated as part of the nominal processing, but it can also be approximated by $\not{p} \approx e^{\left[-\pi W_r^2 (u\xi+v\eta)^2\right]}$, where W_r is the relative bandwidth of the filters (i.e. bandwidth divided by the central frequency)

- (u_{mn}, v_{mn}) are the baseline coordinates in the frequency domain
- (ξ', η') are the coordinates of the resulting distribution in the antenna frame
- d is the antenna element spacing (0.875)
- f_0 is the central frequency (1413 MHz)

The resulting distribution over the antenna frame, cut at half of the maximum (–3dB) will yield a contour (circular or elliptical) that must be projected over the Earth. As mentioned, the major semi-axis may not be oriented along the azimuth (ϕ_p) observation angle, so it should be necessary to project at least six points to compute the ellipse axes on the ground by fitting those points to an ellipse quadratic equation, using a method shown in [RA.02].

As a further simplification, calculating the pixel footprint can be described as the geometrical problem of finding the intersection of an elliptical cone, defined in a satellite frame, with a plane defined in the Earth topocentric frame of the target. The low level description of the algorithm can be found in [RD.13].

Given a looking direction on the satellite, defined by an azimuth (φ) and an elevation (θ), a circumference on the xOy satellite plane (that corresponds to the -3dB contour of the Synthetic Antenna Directional Gain) and the unit semi-sphere of the satellite an elliptical cone can be defined in a satellite reference frame. The projection of its base on the target topocentric plane defines an ellipse.

The following figure depicts the geometry of the problem. The meaning of the reference frames illustrated in red and blue is clarified in Section 5.11.

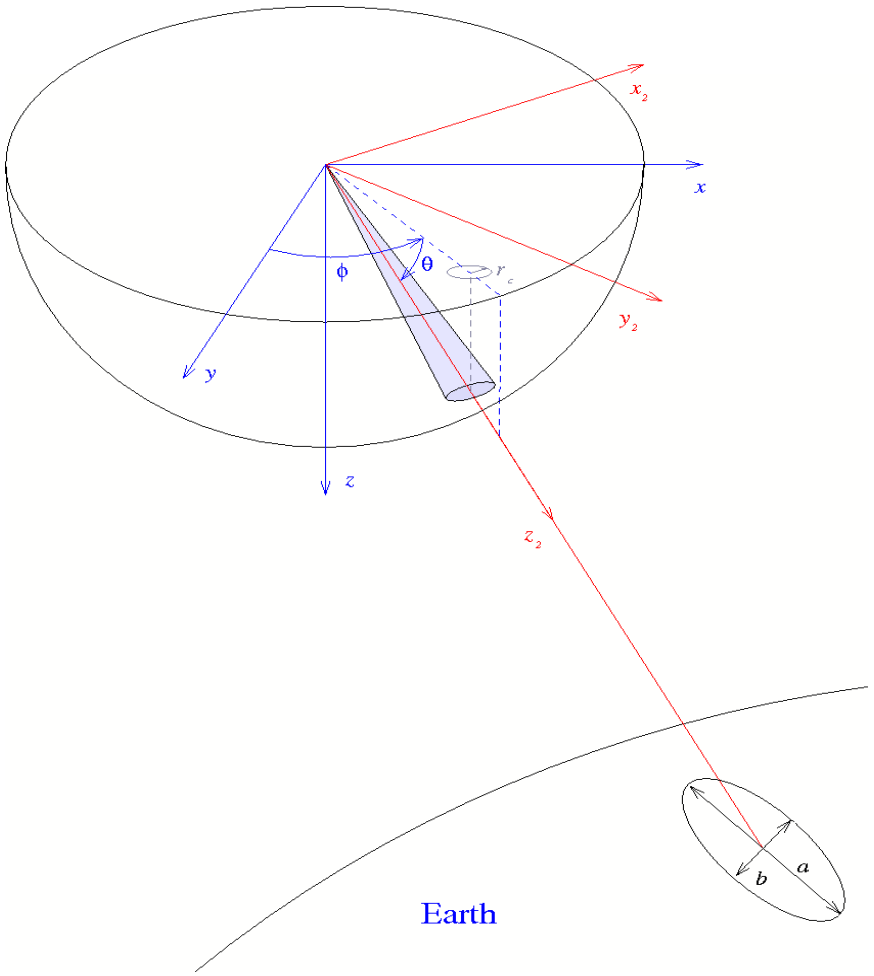

Figure 6: Cone defined on the satellite frame and projection on to the Earth. In blue is the original reference frame on the satellite and in red it is the cone-aligned one. Other quantities are defined in the text.

3.3. Apodisation window computation

3.3.1. Standard apodisation

In the nominal case, the apodisation window is simply a function of the *u,v* baseline coordinates.

For the Blackman window, the expression is the following:

$$W(u,v) = 0.42 + 0.5 \cdot \cos\left(\pi \frac{\sqrt{u^2+v^2}}{\sqrt{3}N_{EL}d}\right) + 0.08 \cdot \cos\left(2\pi \frac{\sqrt{u^2+v^2}}{\sqrt{3}N_{EL}d}\right) \qquad \textit{Eq. 17}$$

In which the MIRAS/SMOS case for a Y-shaped array has already been taken into account, and N_{EL}=21 and d=0.875.

3.3.2. Strip-Adaptive method

In order to ensure a circular footprint on the ground, the process must be initiated backwards, i.e. given the Earth pixel; a ground circular footprint is considered and projected onto the antenna frame as an ellipse. This ellipse shall be the desired output of the –3dB contour for the Array Factor computed in equation 10, which will require a specific window for that purpose.

As seen in [AD.6] the rotation matrix describing the coordinate transformation between Earth fixed grid pixel coordinates and antenna plane spherical coordinates is given by:

$$\begin{pmatrix} dx_{lP} \\ dy_{lP} \\ dz_{lP} \end{pmatrix} = \begin{pmatrix} \cos\varepsilon\cos\phi_g & \cos\varepsilon\sin\phi_g\cos t - \sin\varepsilon\sin t & -\cos\varepsilon\sin\phi_g\sin t - \sin\varepsilon\cos t \\ -\sin\phi_g & \cos\phi_g\cos t & -\cos\phi_g\sin t \\ \sin\varepsilon\cos\phi_g & \sin\varepsilon\sin\phi_g\cos t + \cos\varepsilon\sin t & -\sin\varepsilon\sin\phi_g\sin t + \cos\varepsilon\cos t \end{pmatrix} \times$$
$$\times \begin{pmatrix} r\cos\theta\cos\phi & -r\sin\theta\sin\phi & \sin\theta\cos\phi \\ r\cos\theta\sin\phi & r\sin\theta\cos\phi & \sin\theta\sin\phi \\ -r\sin\theta & 0 & \cos\theta \end{pmatrix} \times \begin{pmatrix} d\theta \\ d\phi \\ dr \end{pmatrix} = [L] \times \begin{pmatrix} d\theta \\ d\phi \\ dr \end{pmatrix} \qquad \textit{Eq. 18}$$

In which

- (x_l, y_l, z_l) are the pixel coordinates in the Earth fixed grid
- ε is the arc between the nadir and the point position from the centre of the Earth assumed as spherical
- ϕ_g is the azimuth angle from the nadir
- t is the tilt angle of the antenna
- r, θ and ϕ are the spherical coordinates of the pixel in the antenna frame
- L is the complete rotation matrix

Projecting a circular pixel of radius R on the Earth surface into the antenna frame using the rotation matrix L, results in an ellipse equation expressed in xi-eta coordinates, as demonstrated in [AD.6].

$$A \cdot \Delta\xi^2 + 2 \cdot B \cdot d\Delta\xi \cdot \Delta\eta + C \cdot \Delta\eta^2 = 1 \qquad \textit{Eq. 19}$$

Where the ellipse coefficients are:

$$A = \frac{1}{R^2}\left((L_{11}^2 + L_{21}^2)\frac{\cos^2\phi}{\cos^2\theta} - 2\cdot(L_{11}L_{12} + L_{21}L_{22})\frac{\cos\phi\sin\phi}{\cos\theta\sin\theta} + (L_{12}^2 + L_{22}^2)\frac{\sin^2\phi}{\sin^2\theta} \right) \qquad \textit{Eq. 20}$$

$$B = \frac{1}{R^2}\left((L_{11}L_{12} + L_{21}L_{22})\frac{2\cos^2\phi - 1}{\cos\theta\sin\theta} - (L_{11}^2 + L_{21}^2)\frac{\cos\phi\sin\phi}{\cos^2\theta} - (L_{12}^2 + L_{22}^2)\frac{\cos\phi\sin\phi}{\sin^2\theta} \right) \qquad \textit{Eq. 21}$$

$$C = \frac{1}{R^2}\left((L_{11}^2 + L_{21}^2)\frac{\sin^2\phi}{\cos^2\theta} + 2\cdot(L_{11}L_{12} + L_{21}L_{22})\frac{\cos\phi\sin\phi}{\cos\theta\sin\theta} + (L_{12}^2 + L_{22}^2)\frac{\cos^2\phi}{\sin^2\theta} \right) \qquad \textit{Eq. 22}$$

The angle of the first semi-axis with respect to the axis ξ is:

$$\delta = \arctan\left(\frac{2B}{A-C}\right) \qquad \textit{Eq. 23}$$

and the semi-axis are defined by:

$$E_1 = \sqrt{A\cos^2\delta + 2B\sin\delta\cos\delta + C\sin^2\delta} \qquad \textit{Eq. 24}$$

$$E_2 = \sqrt{A\sin^2\delta - 2B\sin\delta\cos\delta + C\cos^2\delta} \qquad \textit{Eq. 25}$$

So to circularise each pixel, it is required to compute its elliptical projection in the antenna frame using Eqs. 17 to 19.

Once we have the desired ellipse parameters in the antenna frame, it is required to find a suitable apodisation window whose Equivalent Array Factor –3dB contour fits that ellipse.

For this purpose, a previous study described in [RD.10] shows how to adjust a 2D window based in Kaiser windows. This apodisation window expression is the following:

$$W(u,v) = \frac{I_0\left(\alpha_u\sqrt{1-\left(\frac{u'}{\rho_{\max}}\right)^2}\right)}{I_0(\alpha_u)} \times \frac{I_0\left(\alpha_v\sqrt{1-\left(\frac{v'}{\rho_{\max}}\right)^2}\right)}{I_0(\alpha_v)} \qquad \textit{Eq. 26}$$

Where I_0 is the Modified Bessel Function of the First Kind, and has the following expression:

$$I_0(x) = \sum_{k=0}^{\infty}\frac{(x/2)^{2k}}{(k!)^2} \qquad \textit{Eq. 27}$$

The alpha parameter in each Kaiser window drives the tapering capabilities of the apodisation, meaning that higher values of alpha provide higher tapering. The baselines are normalised with the same maximum value $\rho_{\max} = \sqrt{u_{\max}^2 + v_{\max}^2} = \sqrt{3}N_{EL}d$.

Rotation is introduced by making a linear combination of the u and v frequencies and applying the apodisation along those new directions, as it is shown in the following equation:

$$\begin{aligned} u' &= u\cos\delta + v\sin\delta \\ v' &= -u\sin\delta + v\cos\delta \end{aligned} \qquad \textit{Eq. 28}$$

The problem to be solved would be to find the α_u and α_v coefficients by forcing that the Equivalent Array factor as computed in Eq. 10 particularised at the semi-axes points needs to be half of the maximum (measured at xi=eta=0). The expression has been simplified considering FWF unity on the right side of the equations:

$$\sum_u \sum_v \frac{W(u',v')}{2} = \sum_u \sum_v W(u',v') \times e^{j2\pi u E_1}$$
$$\sum_u \sum_v \frac{W(u',v')}{2} = \sum_u \sum_v W(u',v') \times e^{j2\pi v E_2}$$

Eq. 29

Substituting Eqs. 20 and 22 in the above one, it results in an equation system with two unknowns:

$$\frac{1}{2}\sum_u \sum_v I_0\left(\alpha_u \sqrt{1-\left(\frac{u\cos\delta + v\sin\delta}{\rho_{\max}}\right)^2}\right) \times I_0\left(\alpha_v \sqrt{1-\left(\frac{-u\sin\delta + v\cos\delta}{\rho_{\max}}\right)^2}\right) =$$
$$= \sum_u \sum_v I_0\left(\alpha_u \sqrt{1-\left(\frac{u\cos\delta + v\sin\delta}{\rho_{\max}}\right)^2}\right) \times I_0\left(\alpha_v \sqrt{1-\left(\frac{-u\sin\delta + v\cos\delta}{\rho_{\max}}\right)^2}\right) \times e^{j2\pi u E_1}$$

Eq. 30

$$\frac{1}{2}\sum_u \sum_v I_0\left(\alpha_u \sqrt{1-\left(\frac{u\cos\delta + v\sin\delta}{\rho_{\max}}\right)^2}\right) \times I_0\left(\alpha_v \sqrt{1-\left(\frac{-u\sin\delta + v\cos\delta}{\rho_{\max}}\right)^2}\right) =$$
$$= \sum_u \sum_v I_0\left(\alpha_u \sqrt{1-\left(\frac{u\cos\delta + v\sin\delta}{\rho_{\max}}\right)^2}\right) \times I_0\left(\alpha_v \sqrt{1-\left(\frac{-u\sin\delta + v\cos\delta}{\rho_{\max}}\right)^2}\right) \times e^{j2\pi v E_2}$$

Which unfortunately cannot be solved analytically, so the solution is to compute the Equivalent Array Factor according to Eq. 10 using an array of fixed alpha and δ values. Once the Array Factor is computed, the –3dB contour is fitted with an ellipse, from which the E_1, E_2 parameters can be obtained. This procedure described in Eq.24 needs to be done only once, to obtain the parameter distribution of E_1, E_2 and δ against α_u and α_v.

The way to obtain the corresponding apodisation parameters, required for generating an elliptical footprint in the antenna frame, is then to interpolate in the parameter distribution. Please refer to [AD.6] to find detailed plots showing the procedure. The numeric implementation shall be described in Section 5.4.

A system of analytical equations has been derived in [RD.12] by INETI after implementation of the equations above. The system of equations computes the Kaiser parameters (α_u and α_v) out of the relationship between the ellipse semi-axes in the antenna frame. The coefficients used in that system of equations are described in the next equation and have been set as part of the Strip Adaptive ADF (APOD99):

$$\log 10\left(\frac{\alpha_u}{\alpha_v}\right) = -7.2205E^{-8} + 1.9915\log 10\left(\frac{w_1}{w_2}\right) + 1.0776\left[\log 10\left(\frac{w_1}{w_2}\right)\right]^2 + 0.13022\left[\log 10\left(\frac{w_1}{w_2}\right)\right]^3$$

Eq. 31

$$\log 10(\alpha_u \alpha_v) = 8.4703 + 1.5081\log 10(w_1 w_2) - 0.16293\left[\log 10(w_1 w_2)\right]^2 - 0.016226\left[\log 10(w_1 w_2)\right]^3$$

3.4. RFI Flags Parameters Computation

In the final L1c product, extensive reporting on the RFI presence and estimated influence is provided. There are two sets of RFI flags – at snapshot level (1 value for the whole snapshot) and at pixel level (each pixel reports how it has been affected).

The strategy follows the recommendations in [RD.16] and uses the AUX_RFILST ADF as the reference for the known RFI positions.

At snapshot level, two types of flags can be raised:

- Flag the whole snapshot as corrupted, depending on NIR deviations (inherited from L1a RFI flags)
- Flag depending on the power of the most active pixel in the snapshot

The second snapshot flag is used to detect the presence of RFI contamination even when the source is not observed in the EAFFOV. The star frequencies of the measured image are converted into a complete fundamental hexagon and the highest pixel temperature, excluding a circle of 0.02 radius in the cosine director around the Sun position, is retrieved.

Then, 3 flags at snapshot level will warn the users of presence of an RFI if the temperature is above 300K, 350K or 1000K in the image – the exact values are configurable in the L1OP Configuration File.

The difference of this approach to the current X and Y NIR flags is that the detection of RFI presence is done on the entire hexagon, instead of the L1a antenna temperature measurements. These are more reliable, in the sense that less false alarms will be triggered and the effect of the contamination in brightness temperature will be more consistent within several passes.

At pixel level, and depending on the location and settings for each of the know RFIs in the RFI ADF, we can:

- Flag the pixels contained in a circle of radius n-pixels around the RFI position;
- Flag the pixels that form the tails of the RFI (see Figure 14 for details on the “tail” considered for the direct Sun);
- Also at pixel level, a new RFI contamination flag can be set, depending on the estimated RFI System Response computed in L1a:Flag the pixels that are contaminated with different levels of intensity (by default, 10, 20 and above 30K)

The detailed algorithm to compute the L1c RFI flags is described in Section 5.12.4 and in [RD.16].

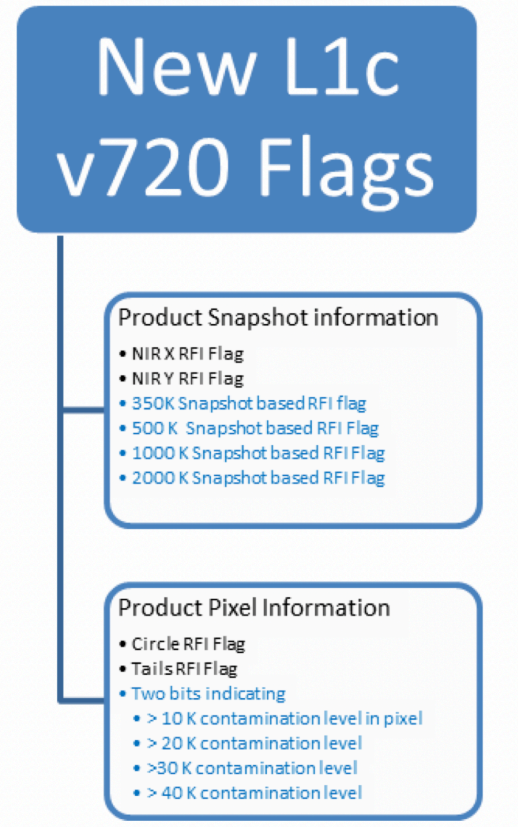

Figure 7: Schematic representation of RFI Flag structure in the L1c products

4. L1B TO L1C HIGH LEVEL DESCRIPTION

The L1b to L1c processing module shall compute the swath ordered brightness temperatures and the Ionospheric Correction factors. L1b data comprises the brightness temperature frequency values. Additional L1a data shall provide the status of the spacecraft (state vector and attitude). Auxiliary data files provide VTEC, Fixed Earth Grid values for lat, long and altitude, geomagnetic model coefficients, RFI mask and specific apodisation window coefficients.

The following figure shows the intended data flow.

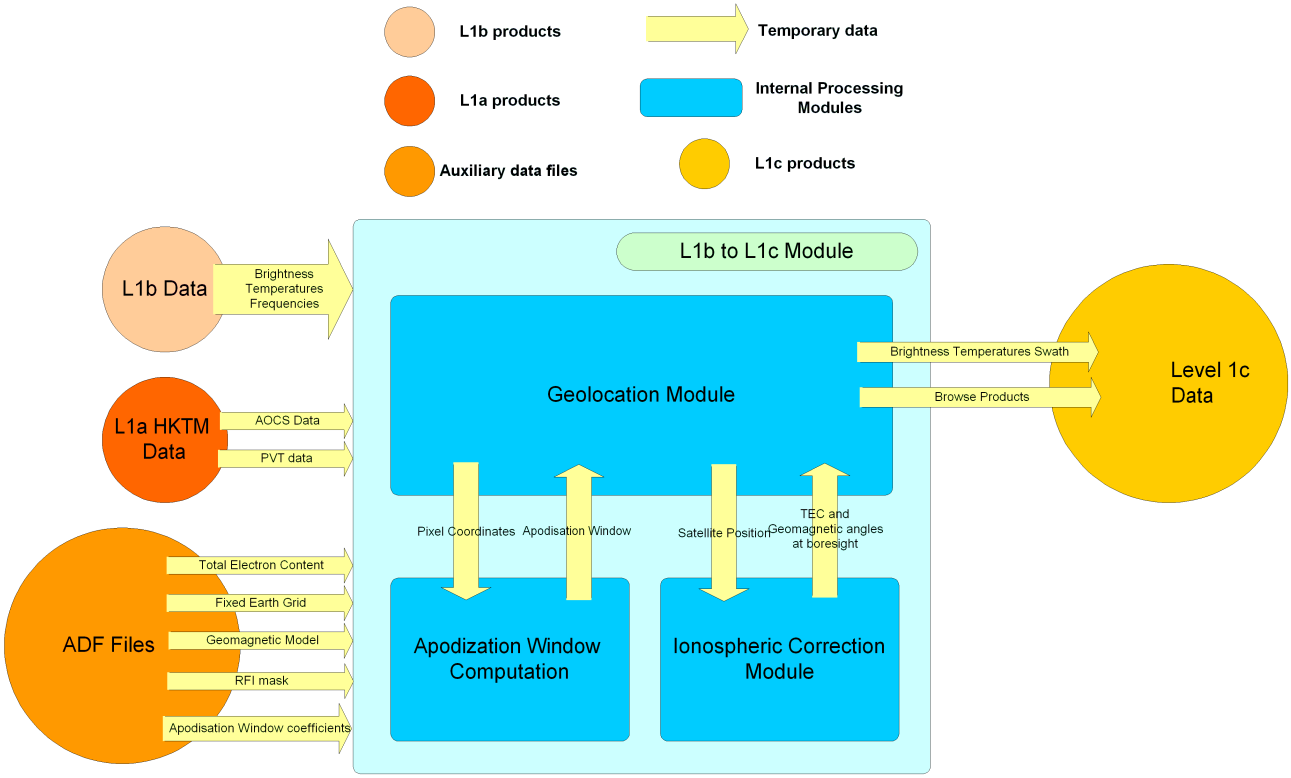

Figure 8: Geolocation Module Data Flow

L1b to L1c processing module will poll, with a predefined interval for the Level 1b products. Every time a new product is detected, the reconstructed data will be forwarded to the geolocation module. The processing is sequential, only the geolocation module needs to be invoked. This module will then process all available data and return the L1c products for storage.

The footprint info shall be attached to each BT value, along with the incidence angle and special flags. The data shall be ordered by pixels, where for each pixel the numeric identifier to the ISEA grid and the number of BT measurements available shall be indicated. Each of the measurements will be composed of the BT component measured (real for H and V, complex for HV), incidence (θ_p) and azimuth (ϕ_p) observation angles, radiometric accuracy, Faraday rotation angle (ω), geometric rotation angle (φ), pixel footprint elliptical major and minor semi-axes and quality flags. These flags shall report the position of the pixel within the FOV (in/out of alias-free FOV, in/out of extended alias-free FOV and close to the EAF-FOV border or to the unit circle border aliases, also known as “suspenders and belts”) as well as

represent any external source that may affect the pixel, like a direct Sun alias, reflected Sun alias, RFI sources, etc. Input data for the Level 2 processing shall always be a geo-sorted L1c product.

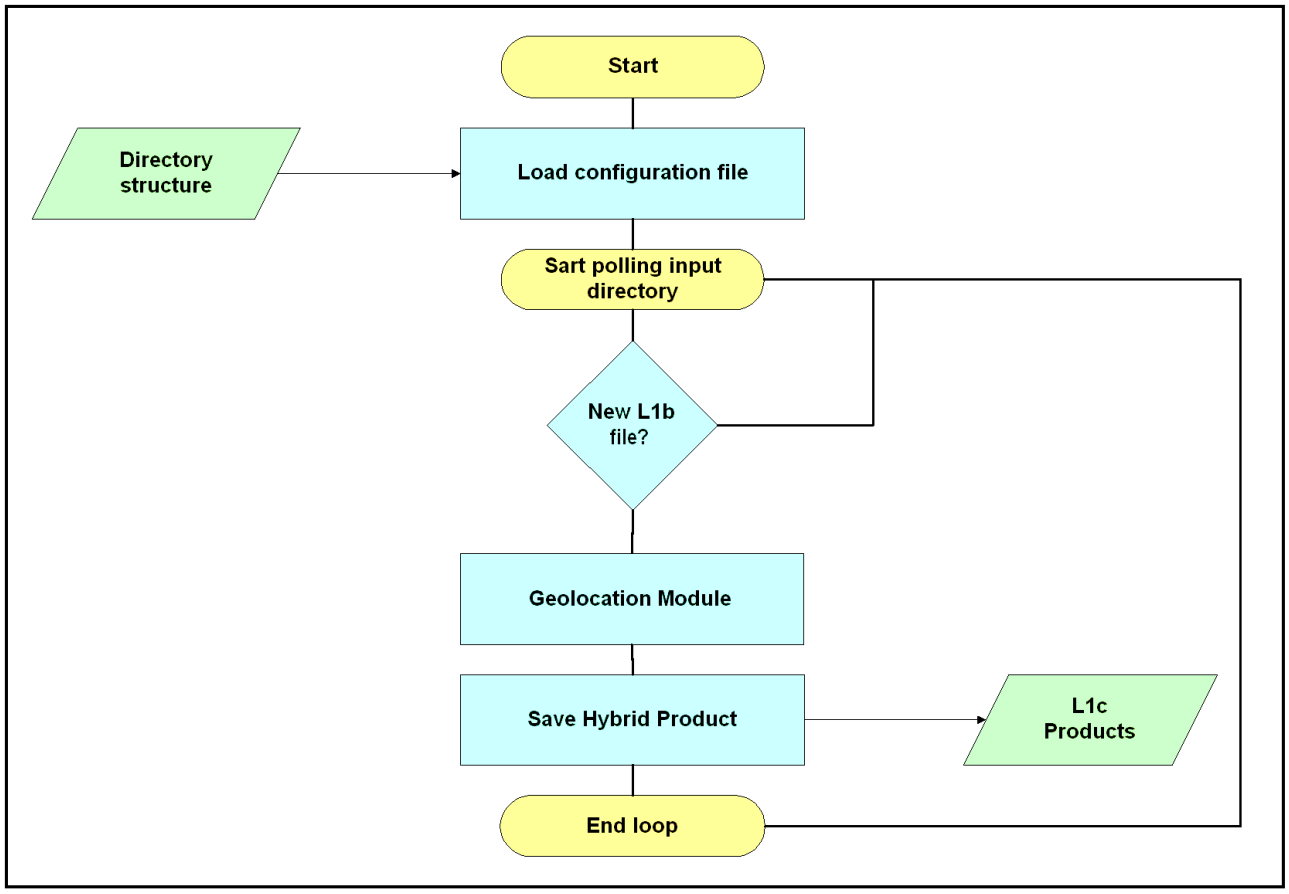

Figure 9: Orchestrator Processing Steps

5. DETAILED PROCESSING MODEL

The following sections will detail the implementation of each processing unit, with processing diagrams and variable lists.

5.1. Geolocation module

The objective of the Geolocation process is first to compute for each scene or integration time the list of Earth fixed pixels falling inside the Extended Alias-Free FOV. For each of these pixels observed, it shall then compute the corresponding Brightness Temperature and observation angles, the radiometric accuracy, the Faraday and geometric rotation angles, the footprint major and minor semi-axis and a set of quality flags corresponding to that particular time.

The output L1c product pixels will then be formed by aggregations of BT values and their incidence observation angles for each pixel covered in that swath. Additionally, there shall also be particular information about the footprint shape and orientation and the snapshot in which the values were measured.

Using the snapshot reference, there shall be another block of data containing relevant information about the snapshot, like UTC time, PVT and AOCS, as well as the TEC and geomagnetic model values used in the Faraday rotation correction.

Different L1c products are generated depending on the polarisation mode of the instrument (dual or full polarimetric) and with the apodisation window used in the processing (either land or sea).

There shall also be Browse L1c products, containing the same information as nominal L1c, but providing only one value per pixel considered at an incidence angle of 42.5º.

The processing diagram explains how the process is done, first at the level of the l1b scenes, and later at the level of each pixel.

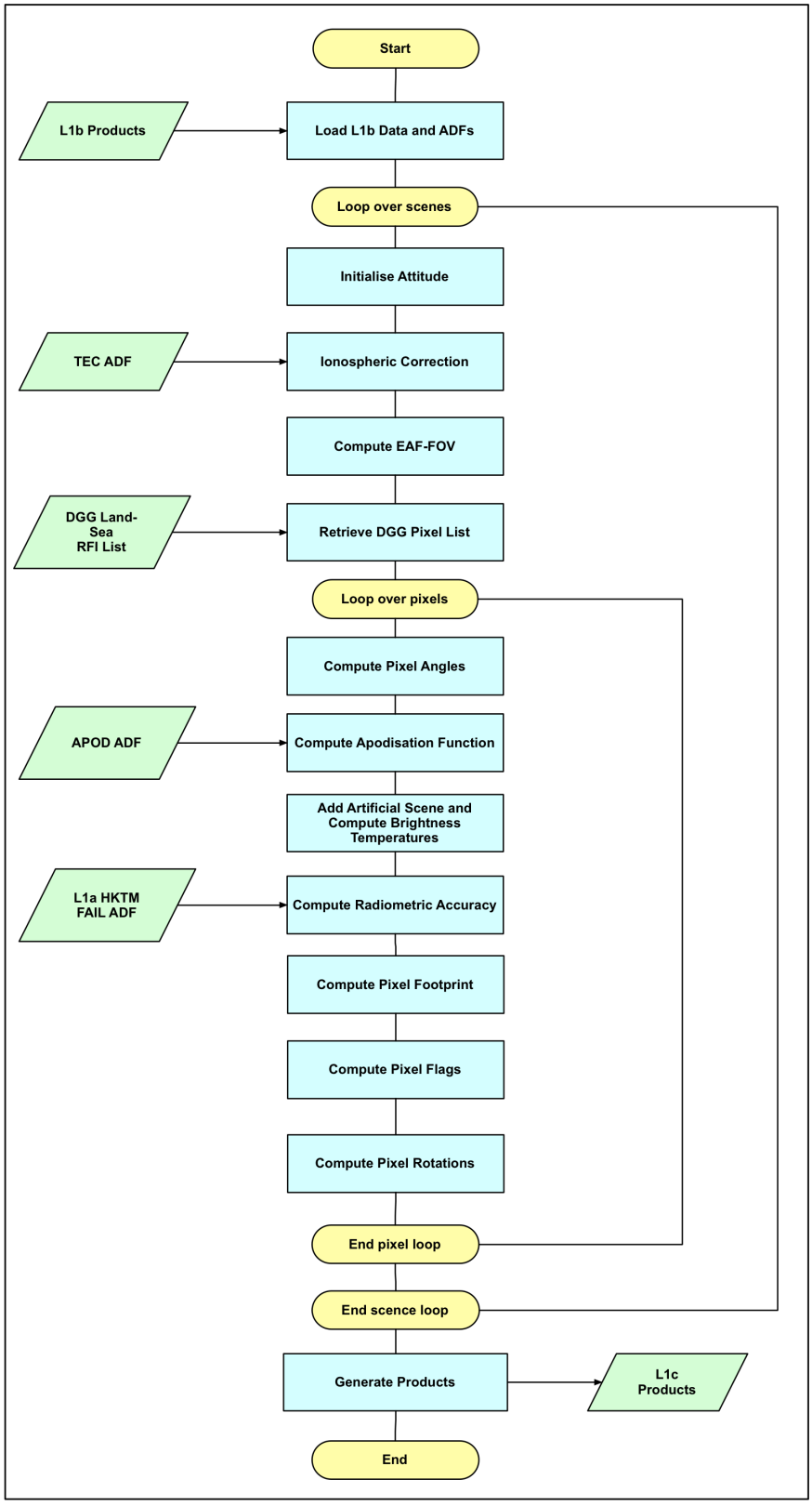

Figure 10: Geolocation module processing steps

5.1.1. Inputs

- Direct interface (since merge of L1b and L1c processors in L1OP v710)
  - Reconstructed $\hat{T}_B$ frequencies at top of the atmosphere
  - Frequencies of the artificial scene (Gibbs 2 mode only)
  - RFI System Response frequencies
  - UTC time
  - PVT and AOCS
  - System Temperatures
- Auxiliary Data
  - Earth Fixed Grid (SM_xxxx_AUX_DGG___<ID>)
  - Land Sea Mask (SM_xxxx_AUX_MASK__<ID>)
  - RFI mask (SM_xxxx_AUX_RFI___<ID>)
  - Average Antenna Patterns (SM_xxxx_AUX_PATT<ID>)
  - Best Fit Plane (SM_xxxx_AUX_BFP___<ID>)
  - Mispointing Angles (SM_xxxx_AUX_MISP__<ID>)
  - Default PLM values (SM_xxxx_AUX_PLM___<ID>)

5.1.2. Outputs

- L1c data (T_B swath)
  - Science Data, Land, Dual Polarisation (SM_xxxx_MIR_SCLD1C<ID>)
  - Science Data, Land, Full Polarisation (SM_xxxx_MIR_SCLF1C<ID>)
  - Science Data, Sea, Dual Polarisation (SM_xxxx_MIR_SCSD1C<ID>)
  - Science Data, Sea, Full Polarisation (SM_xxxx_MIR_SCSF1C<ID>)
  - Browse Data, Land, Dual Polarisation (SM_xxxx_MIR_BWLD1C<ID>)
  - Browse Data, Land, Full Polarisation (SM_xxxx_MIR_BWLF1C<ID>)
  - Browse Data, Sea, Dual Polarisation (SM_xxxx_MIR_BWSD1C<ID>)
  - Browse Data, Sea, Full Polarisation (SM_xxxx_MIR_BWSF1C<ID>)

5.1.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
filename	Name of L1b science data file to be processed	N/A	char	I	N/A	pointer
aocs_data	Structure with processed AOCS data	N/A	t_l1a_aocs	I	N/A	1
pvt_data	Processed PVT data	N/A	t_l1a_pvt	I	N/A	1
l1b_data	L1b data	Section 3.2.2	t_l1b_bt_freq	I	N/A	1
l1b_rfi_freqs_data	RFI System Response in frequency domain	[AD.8]	double	I	N/A	2791
l1b_temps_freqs_data	Artificial Gibbs 2 scene in frequency domain	[AD.8]	double	I	N/A	11164
earth_grid	Earth Fixed Grid	Section 3.2.1	double	I	N/A	4x2.6x10^6
rfi_mask	RFI mask, 0- no RFI, 1 – RFI present	Section 3.2.1	bool	I	N/A	2.6x10^6
earth_af_fov	AF-FOV border projection into Earth grid coordinates	Section 3.2.1	t_l1c_af_fov_earth	L	N/A	Scene dependant
earth_pixel_list	List of pixels inside AF-FOV border projection into Earth grid coordinates	Section 3.2.1	t_l1c_af_fov_list	L	N/A	Scene dependant
u_v	U/V coordinates of non-redundant baselines	[RD.17]	t_l1c_baselines	L	N/A	NON_RED_BASELINES
antenna_patt	Averaged LICEF receiver directional power Gain	Section 3.2.3	double_complex	I	N/A	1
directivity	Averaged directivity of the LICEFs	Section 3.2.3	double	I	N/A	1
sys_temp	Averaged System Temperature measured by the PMS system	[AD.8]	double	I	Kelvin	1
l1c_data	L1c data per pixel	Section 4	t_l1c_data	O	Kelvin	1

Variable name	Description	Definition	Type	Class	Unit	Size
bt_accuracy	Radiometric Accuracy	Eq. 15	double	O	Kelvin	1
ionospheric_rotation_angles	Ionospheric Rotation Angles	Eq. 1	double	O	degrees	2
footprint_size	Equivalent Array Factor	Eq. 16	double	O	meters	2
pixel_observation_angles	Pixel observation angles	Section 3.1.3	t_l1c_obs_angles	O	degrees	2

Table 4: Geolocation module variable list

5.1.4. Implementation

The geolocation processing shall ingest a L1b product file, and for each scene contained inside it shall compute a list of Earth Fixed pixels observed, along with the magnitudes being measured (Brightness Temperature, radiometric accuracy, observation angles, rotation angles, footprint size and quality flags)

The process, as indicated on figure 9, must first perform a sequence of computations valid for each scene, and later on another sequence of computations for each pixel contained inside the scene. Details are given on each sub-process in the next sections, using one section per box in figure 9.

Basically, the approach to be taken is to process each L1b scene at a time, computing the Earth Fixed pixels contained inside, and then for each pixel compute the corresponding output values. In the nominal case, most of the pixels computed for one scene should have been also computed in the previous one, with the exception of some new ones appearing and others no longer visible as the S/C moves between scenes. The end result shall be a list of pixels (around 80000 different ones for a 2500 scene product), and for each pixel there shall be a list of associated measurements, taken as it traverses the FOV from the top to the bottom.

What shall be described here is the consolidation approach to be taken, in order to store in L1c products only pixels whose measurement lists are complete.

This can be achieved by first limiting the output of the product by taking out all pixels contained in the last scene set in the L1b product. This ensures that all pixels to be written in the L1c product shall contain the whole list of measurements (i.e. observation angles). A good measure is to control this behaviour through a configuration flag in the L1c processing.

A second step needed is to process this L1b product together with the previous L1b product. This allows introducing in the product the pixels belonging to the last scene of the previous L1b product, and all their measurements, which shall complement the measurements available from the current L1b product processing.

In order to achieve this, the processing of this previous product must be performed “backwards”, starting from the last L1b scene, and then progressing to previous scenes, but only using the pixels filtered from the last scene. This will insert into the L1c product the pixels and measurements that were removed when the previous product was processed.

A final step is to extract the brightness temperatures for pixels with an observation angle of 42.5º and create a browse field, also part of the l1c_data product.

5.1.5. Error Handling

Error Nr	Error Description	Error Code Returned	Program behaviour	Quality flag raised
1	File accessor unable to read the L1b file	DATA_CORRUPTED	Processing unit returns with no data processed.	No

Table 5: Geolocation module error handling

5.2. Initialise Attitude

This computation initialises all the EE CFI required structures that shall be used in later functions. These structures are composed by a *time_id* and an *attitude_id* structure.

The *time_id* shall be used in all time conversions and coordinate system change. It can be initialised at the beginning of processing. However, the *attitude_id* structure must be computed for each scene, as it represents the pointing angles for the instrument at a particular time.

5.2.1. Inputs

- ❑ Direct interface (since merge of L1b and L1c processors in L1OP v710)
  - UTC time
  - PVT and AOCS
- ❑ Auxiliary Data
  - Time Correlations (SM_xxxx_MPL_ORBSCT<ID>)
  - Bulletin B ((SM_xxxx_AUX_BULL_B_<ID>)
  - Best Fit Plane (SM_xxxx_AUX_BFP___<ID>)
  - Mispointing Angles (SM_xxxx_AUX_MISP__<ID>)

5.2.2. Outputs

- ❑ EE CFI structure time_id
- ❑ EE CFI structure attitude_id
- ❑ matrix_BFP2EF

5.2.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
pvt_data	Processed PVT data	N/A	t_l1a_pvt	I	N/A	1
utc_time	UTC snapshot time	N/A	double	I	days	1
aocs_data	Processed AOCS data	N/A	t_l1a_aocs	I	N/A	1
BF_to_BFP_matrix	Body Frame to Best Fit Plane rotation matrix	N/A	double	I	N/A	3x3
STR_to_BF_matrix	Star Tracker to Body Frame rotation matrix	N/A	double	I	N/A	3x3
time_id	EE CFI time structure	N/A	xl_time_id	O	N/A	1
attitude_id	EE CFI pointing structure	N/A	xp_attitude_id	O	N/A	1
matrix_BFP2EF	Rotation matrix between the BFP and the EF frame	N/A	double	O	N/A	9

Table 6: Initialise Attitude variable list

5.2.4. Implementation

First of all, the time_id structure must be initialised using time correlations available in any reference file. Currently the most recent IERS bulletin B is used as input auxiliary file, but in case it is not available, an older bulletin B or even the Orbit Reference File can be used to retrieve these values.

Once this correlations are obtained, the time_id structure is initialised using the EE CFI function *xl_time_ref_init* using as input those correlations and receiving the *time_id* structure.

Once a valid time_id structure is available, the *attitude_id* must be initialised using the scene time, PVT and AOCS data. The initialisation is done using EE CFI function:

- xp_attitude_user_set(&time_id, attitude_id, /* input / output */
  &time_reference, &time, pvt_data.position, pvt_data.velocity,
  acceleration, &target_frame, TOD_to_BFP_matrix, matrix_rate, matrix_rate_rate,
  offset, ierr_xp)

The inputs for this function are the following:

1. time_id (previously computed)
2. time_reference (set to XL_TIME_UTC)
3. time is the scene time expressed in decimal days since 1st Jan 2000 00:00:00
4. pvt_data.position
5. pvt_data.velocity
6. acceleration (set to [0, 0, 0])
7. target_frame (set to XP_INSTR_ATT)
8. TOD_to_BFP_matrix (computed in next step)
9. matrix_rate (set to [0, 0, 0, 0, 0, 0, 0, 0, 0])
10. matrix_rate_rate (set to [0, 0, 0, 0, 0, 0, 0, 0, 0])
11. offset (set to [0, 0, 0])

The output shall be the attitude_id structure already initialised.

The matrix TOD_to_BFP_matrix (True Of Date to Best Fit Plane) shall be computed by the multiplication of three other matrices: STR_BF_matrix (Star Tracker to Body Frame), BF_BFP_matrix (Body Frame to Best Fit Plane) and TOD_to_BF_matrix (True Of Date to Body Frame), described below.

The BF to BFP rotation matrix takes in account the antenna phase centres of the MIRAS array, and corresponds to a rotation from the Body Frame with Euler angles yaw (Ψ), pitch (Θ), and roll (Φ) to the Best Fit Plane. These values are taken from the Best Fit Plane ADF by combining the two rotations contained inside this file (on-ground measured BFP rotation and in-orbit calibrated BFP rotation).

On an equivalent approach, the STR to BF rotation matrix takes into account the possible mispointing that can be accumulated between the Star Tracker and the Antenna frame due to deformations of the payload structure. At launch it will be initialised to zero, but may be modified in flight if such deformations are detected. This STR to BF rotation also corresponds to a rotation from the Star Tracker Frame with Euler angles yaw (Ψ), pitch (Θ), and roll (Φ) to the Body Frame. These values are taken from the Mispointing ADF by combining the three rotations contained inside this file (Proteus to Body rotation, Body to Antenna rotation and in-orbit calibrated Mispointing rotation).

Both matrices are constructed by multiplying, in the inverse order, the rotation matrices for each Euler angle, resulting in the following matrix:

$$\begin{bmatrix} \cos\Phi\cos\Theta & \cos\Phi\sin\Theta\cos\Psi+\sin\Phi\sin\Psi & \cos\Phi\sin\Theta\sin\Psi-\sin\Phi\cos\Psi \\ -\sin\Theta & \cos\Theta\cos\Psi & \cos\Theta\sin\Psi \\ \sin\Phi\cos\Theta & \sin\Phi\sin\Theta\cos\Psi-\cos\Phi\sin\Psi & \sin\Phi\sin\Theta\sin\Psi+\cos\Phi\cos\Psi \end{bmatrix} \qquad \textit{Eq. 32}$$

In order to compute the matrix TOD_to_BF_matrix (True Of Date to Body Frame), two other matrices must be computed and multiplied.

First the matrix J2000 to Body Frame is computed, by converting the quaternions in the aocs_data structure into vectors. This is done by using function:

- xl_quaternions_to_vectors (quaternions, ux_vec, uy_vec, uz_vec, ierr_xl)

Special care must be taken with the quaternion ordering, as for SMOS data the real element is the first element, whereas in this CFI function it must be the last. The resulting vectors set in a matrix and with negative sign represent the J2000 to BF rotation matrix. The negative sign is needed due to different conventions between SMOS and CFI functions

The second matrix to be computed is the TOD to J2000, which is obtained by transforming the three unitary vectors from TOD to GM2000 using CFI function:

- xl_change_cart_cs(&time_id, &mode, &cs_in, &cs_out, &time_reference, &time, pos_in, vel, acc, pos_out, vel_out, acc_out)

Where the input variables are:

1. cs_in (set to XL_TOD)
2. cs_out (set to XL_GM2000)
3. pos_in (set alternatively to [1, 0, 0], [0, 1, 0] and [0, 0, 1])
4. vel and acc (set to [0, 0, 0])

The three pos_out vectors, set in matrix form, compose the TOD to J2000 rotation matrix.

These four matrices must be now multiplied in order, and as a final step, in the resulting matrix the first line must be swapped with the last line in order to cope with the different axis convention between SMOS and CFI.

$$\begin{cases} X_{SMOS} = -Z_{CFI} \\ Y_{SMOS} = -Y_{CFI} \\ Z_{SMOS} = -X_{CFI} \end{cases} \qquad \textit{Eq. 33}$$

Additionally, from the TOD_to_BF_matrix a new rotation matrix is created (matrix_BFP2EF) using the same CFI function as before but with input variables:

1. cs_in (set to TRUE_OF_DATE)
2. cs_out (set to EARTH_FIXED)
3. pos_in (columns of ToD_to_BF_matrix)
4. vel and acc (set to [0, 0, 0])

This rotation matrix is used in other routines of L1OP, such as the Pixel Footprint calculation (Section 5.11)

5.2.5. Error Handling

In case of time_id initialisation failure, the processing of the L1b product must be aborted. There can be no time conversions without a valid time_id.

In case of attitude_id initialisation failure, the affected scene must be skipped, but the process of the remaining scenes can continue. There cannot be any pixel geolocation without a correct attitude_id valid for each scene.

5.3. Ionospheric Correction module

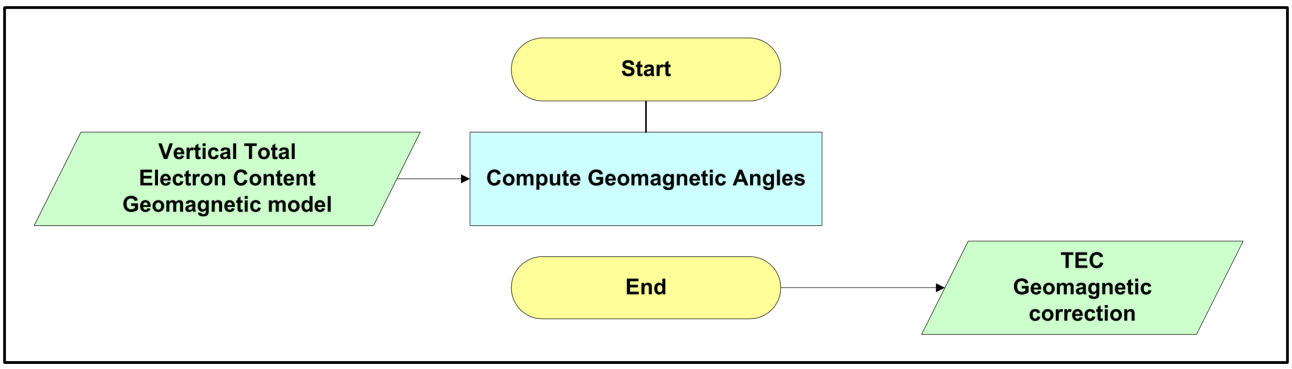

Figure 11: Ionospheric Correction module processing steps

The Ionospheric Correction module computes the geomagnetic field parameters based on the International Geomagnetic Reference Field model (IGRF2015). The Total Electron Content is also computed, based on Total Electron Content (TEC) ADF maps or on the default IRI2001 model.

5.3.1. Inputs

- Direct interface (since merge of L1b and L1c processors in L1OP v710)
  - UTC time
  - PVT Data
- Auxiliary Data
  - Vertical Total Electron Content Data (SM_xxxx_AUX_VTECy_<ID>, with y being P/R/C)
  - Geomagnetic Model Data file available with the IGRF 12th generation model

5.3.2. Outputs

- Total Electron Content Data (TEC)
- Magnetic Field Strength (F)
- Magnetic Field inclination (I) and declination (D)

5.3.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
sc_position	Processed PVT data	N/A	double	I	m	3

Variable name	Description	Definition	Type	Class	Unit	Size
utc_time	UTC snapshot time	N/A	double	I	days	1
geo_results	Vector to hold full IGRF2015 model output	N/A	double	L	N/A	8
dyear	UTC snapshot time expressed in decimal years	N/A	double	I	year	1
x_lat	IRI2001/ IGRF2015 interface, Satellite latitude	Geodetic latitude	double	L	degrees	1
x_long	IRI2001/ IGRF2015 interface , Satellite longitude	Geodetic longitude	double	L	degrees	1
x_height	IRI2001/IGRF2015 interface, Satellite altitude	Altitude	double	L	m	1
jf	IRI2001 interface, model options	IRI2001 function header	int	L	N/A	30
jmag	IRI2001 interface, optional input parameters	IRI2001 function header	int	L	N/A	1
year	IRI2001 interface, integer year from UTC time	IRI2001 function header	int	L	years	1
mmdd	IRI2001 interface, integer month and day from UTC time (MMDD)	IRI2001 function header	int	L	N/A	1
hour	IRI2001 interface, decimal hour from UTC time	IRI2001 function header	float	L	hours	1
htecmin	IRI2001 interface, minimum height for TEC integration	IRI2001 function header	int	L	km	1
geomag_f	Magnetic field strength	Section 3.1.1	double	O	Tesla	1
geomag_d	Magnetic field declination	Section 3.1.1	double	O	degrees	1

Variable name	Description	Definition	Type	Class	Unit	Size
geomag_i	Magnetic field inclination	Section 3.1.1	double	O	degrees	1
tec	Total Electron Content	Section 3.1.1	double	O	TEC	1

Table 7: Ionospheric Correction variable list

5.3.4. Implementation

The geomagnetic model is available from the Fortran model IGRF2015 model, as explained in Section 3.1.1, and a suitable C wrapper must be implemented to interface it. First, the relevant satellite data must be computed from the UTC time and PVT data retrieved for every snapshot processed:

1. UTC time shall be converted from its input format into decimal years
2. PVT position data shall be converted to geodetic latitude and longitude using CFI function *xl_cart_to_geod*

The IGRF2015 baseline model shall be the C model available in http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html, and available at 22st December 2014. This C model contains more expansion coefficients beyond 2010 than the fortran counterpart.

The IGRF2015 model requires these three inputs above (dyear, x_lat and x_long), plus the altitude at which the geomagnetic field will be computed. In this case, the altitude desired (x_height) shall always be 400km:

```
double geo_results[8];   // Vector to hold full Igrf2000 model output


//Compute the geomagnetic field, based in the International Geomagnetic
//Reference Field model. It is implemented in a set of C functions, obtained
//from the IAGA site. These can be found at
//http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html


error = computeIgrf2000(dyear, x_height, x_lat, x_long, geo_results[0], geo_results[1],
geo_results[2],    geo_results[3],    geo_results[4],    geo_results[5],    geo_results[6],
geo_results[7]);
If(error != 0)
     Error #1 handled
// We are only interested three parameters from the eight returned by Igrf2000 routine


geomag_f = geo_results[0];   // babs field in Igrf2000 interface
geomag_i = geo_results[4];   // dip field in Igrf2000 interface
geomag_d = geo_results[5];   // dec field in Igrf2000 interface
```

The resulting variables are **geomag_f** (magnetic field strength), **geomag_d** (magnetic field declination), and **geomag_i** (magnetic field inclination), all values at satellite height.

Next, the TEC value applicable to that position and time can be computed from the available ADF by interpolation, or using the IRI2001 model in case there are no files available:

In case of interpolation using ADF, the files used are the IGS combined, which contain data on a mercator grid and a temporal sampling of 2 hours. Each file contains 13 layers, being the first layer a repetition of the last layer from the previous file.

Thus, the interpolation shall be first performed as a spline (cubic) interpolation in time, taking into account the UTC time requested and the UTC times of the datasets (e.g. possibly generating an intermediate layer over the whole mercator grid). The spline takes into account the 13 layers used, so that the complete behaviour of the TEC field is captured over 24h.

Afterwards, a simple bi-linear interpolation is required in lat-lon (interpolating between the four closest grid points for each lat-lon point). A note of advice is mentioned here, as the data in the IGS combined is expressed in 0.1 TEC Units.

If the option to compute the TEC value using the IRI2001 model is selected, then this FORTRAN model (in particular function *irit13ne*) must be invoked using the following parameters:

1. x_lat and x_long as computed before from PVT position data
2. jmag option as 0.
3. jf model options vector as all 1, except for positions 5, 6 and 21 which should be 0.
4. UTC time converted into Local Time and expressed as integer year, integer month+day (MMDD) and decimal hour (conversion from UTC into Local Time is achieved by adding 25 hours to the decimal hours)

5. start and stop altitudes, where the start altitude is selected as 50km and stop altitude is 400km.

The output of this function shall be the TEC value expressed in TEC Units (1 TECU = 1e+016e^-/m^2)

The IRI2001 baseline model shall be the fortran model available in ftp://nssdcftp.gsfc.nasa.gov/models/ionospheric/iri/iri2001/fortran_code, and available at 31st October 2007.

In either case, the resulting value, `tec`, is the Total Electron Content for all pixels in the snapshot. The module shall return `geo_f`, `geo_d`, `geo_i` and `tec` as output for the input UTC time and PVT position.

5.3.5. Error Handling

Error Nr	Error Description	Error Code Returned	Program behaviour	Quality flag raised
1	IGRF2015 model returns error	DATA_CORRUPTED	Processing unit sets default values for geomagnetic data and continues.	Yes
2	File accessor unable to read the VTEC AUX file	DATA_CORRUPTED	Processing unit sets default values for TEC data and returns. Use of IRI2001 model should be tried next	Yes
3	IRI2001 model returns error	DATA_CORRUPTED	Processing unit sets default values for TEC data and returns.	Yes

Table 8: Ionospheric Correction module error handling

5.4. Compute EAF-FOV

This algorithm computes for each scene a discrete list of xi-eta points in the antenna frame, depicting the Extended Alias-Free FOV. It shall also compute two important angles for the spacecraft, which shall be used in further computations: the tilt angle and the local_north angle.

The first computation of the EAF-FOV shall make use of the initialised attitude_id structure only, whereas the angles computation shall also use the pvt_data structure from each scene, containing position and velocity.

5.4.1. Inputs

- attitude_id
- pvt_data

5.4.2. Outputs

- eaf_fov_list
- tilt_angle

- local_north_angle

5.4.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
pvt_data	Processed PVT data	N/A	t_l1a_pvt	I	N/A	1
attitude_id	EE CFI pointing structure	Section 5.2.3	xp_attitude_id	I	N/A	1
eaf_fov_list	List of xi-eta points in the border of the EAF-FOV	N/A		O	N/A	1
tilt_angle	Angle between S/C to Nadir direction and Instrument boresight	N/A	double	O	deg	1
local_north_angle	Angle between S/C ground track instantaneous direction and local north	N/A	double	O	deg	1

Table 9: Compute EAF-FOV variable list

5.4.4. Implementation

The computation of the EAF-FOV is done through the selection of a geometrical boundary that follows certain requirements. These requirements can be observed in figure 6, in section 3.21.

The area must be limited in the first place by a hexagon of radius $\frac{2}{3d}$ (where d is 0.875), and also by the Earth aliases in all six adjacent aliases.

Thus, one of the first tasks is to obtain the discrete contour of the Earth within the unit circle. This contour is obtained by projecting the azimuth and elevation coordinates of the Earth tangent points into the cosines domain. This step needs to be done using EE CFI functions, and is composed of the following steps:

1. Establish the discrete list of azimuth directions, for example from 0º to 360º in incremental steps of 1º
2. For each azimuth direction, call xp_target_altitude using attitude_id structure and the azimuth as input, in order to obtain an initialised target_id structure. Parameter dem_id and atmos_id shall be left NULL, deriv shall be XP_NO_DER.

3. Call xp_target_extra_main with the target_id structure obtained in the previous step, select choice XP_TARG_EXTRA_MAIN_SAT2TARG_ATTITUDE, target_number shall be the num_user_target returned from the previous function and target_type XP_LOS_TARGET_TYPE, in order to obtain several output vectors (main_results, main_results_rate, main_results_rate_rate)
4. The azimuth angle in attitude frame is contained in main_results[6] (ϕ), and the elevation in main_results[7] (θ). Taking into account the CFI angle convention, the cosines domain xi-eta coordinates are computed according to equation 7
5. Gather all xi-eta pairs into a list (eaf_fov_list) to be used later

In order to shift these coordinates to each of the aliases, it must be known that the centers of the aliases are located at:

$$\left.\begin{aligned} xi &= \frac{2}{d\sqrt{3}}\cos(\alpha) \\ eta &= \frac{2}{d\sqrt{3}}\sin(\alpha) \end{aligned}\right\} \text{with } \alpha = \frac{\pi}{6}, \frac{\pi}{2}, \frac{5\pi}{6}, \frac{7\pi}{6}, \frac{3\pi}{2}, \frac{11\pi}{6} \qquad \textit{Eq. 34}$$

With all these inputs known, the selection of the contour passes by establishing the geometrical requirements in an ad-hoc algorithm. For example, the algorithm implemented in L1OP travels trough a discrete list of xi-eta points of the hexagon, and for any given xi value, it selects the appropriate eta value from between the eta value of the hexagon and of the six aliases.

Additionally, the other two angles needed as output are obtained through a series of calls to EE CFI functions. The only inputs required are the attitude_id and the pvt_data structure.

The processing steps are the following:

1. Call xl_cart_to_geod to transform the pvt_data into subsatellite point lat-lon-alt coordinates
2. Use the previous lat-lon coordinates, together with altitude 0 in function xl_geod_to_cart to obtain the position and velocity coordinates of the subsatellite point.
3. Call xp_target_generic with the attitude_id structure plus the position and velocity values of the subsatellite point, in order to obtain an initialised target_id structure. Parameter dem_id shall be left NULL, deriv XP_NO_DER
4. Call xp_target_extra_main with the target_id structure obtained in the previous step, select choice XP_TARG_EXTRA_MAIN_SAT2TARG_ATTITUDE, target_number shall be the num_user_target returned from the previous function and target_type XP_LOS_TARGET_TYPE, in order to obtain several output vectors (main_results, main_results_rate, main_results_rate_rate)
5. The tilt angle is computed from element number 8 in the main_results vector (tilt = 90 – main_results[7])
6. Call xp_target_extra_aux with the target_id structure obtained in the step 3, select choice XP_TARG_EXTRA_AUX_TARGET_NADIR_VEL, target_number shall be the num_user_target returned from the previous function and target_type XP_LOS_TARGET_TYPE, in order to obtain several output vectors (aux_results, aux_results_rate, aux_results_rate_rate)
7. The local north angle is computed from element number 11 in the aux_results vector (aux_results[10]), which represents the azimuth of nadir velocity with respect to local north

5.4.5. Error Handling

In case any error happens in the contour computaion, the processing of the related scene must be aborted. Without the xi-eta contour the DGG list of points cannot be extracted.

In case the error happens during the computation of the tilt, the geometrical rotation of all pixels in that scene cannot be computed.

In case the error happens during the computation of the local north angle, the geometrical rotation and the footprint of the pixels cannot be computed.

5.5. Retrieve DGG pixel list

This algorithm shall filter out the list of Earth fixed pixels available from the ADF according to the contour computed in the previous step. In order to this, first the points in the contour list must be projected onto the Earth's surface. Afterwards, all pixels in the ADF shall be compared against the contour and only those falling INSIDE the contour shall be selected for the scene.

It is advisable at this stage to ingest not only the DGG ADF, but also the Land-Sea mask and the RFI mask, as it is a good point to associate each of the final selected pixels to their corresponding flags.

5.5.1. Inputs

- attitude_id
- eaf_fov_list
- Auxiliary Data
  - Earth Fixed Grid (SM_xxxx_AUX_DGG___<ID>)
  - Land Sea Mask (SM_xxxx_AUX_MASK__<ID>)
  - RFI mask (SM_xxxx_AUX_RFI___<ID>)

5.5.2. Outputs

- earth_pixel_list

5.5.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
attitude_id	EE CFI pointing structure	Section 5.2.3	xp_attitude_id	I	N/A	1
eaf_fov_list	List of xi-eta points in the border of the EAF-FOV	Section 5.4.3		I	N/A	1

Variable name	Description	Definition	Type	Class	Unit	Size
earth_pixel_list	List of pixels from the DGG ADF contained inside the EAF-FOV border	N/A		O	N/A	1

Table 10: Compute DGG pixels variable list

5.5.4. Implementation

First of all, each of the xi-eta points in the contour list of the EAF-FOV must be projected onto the Earth. This is done through the use of several EE CFI functions as in the following approach:

1. For each xi-eta point extract the corresponding azimuth and elevation angles. Azimuth must be converted to [0,360] range before being used by CFI functions.
2. Call xp_target_inter function with attitude_id structure, the azimuth and elevation angles, deriv XP_NO_DER and inter_flag XP_INTER_1ST, in order to obtain an initialised target_id structure. Parameter dem_id and atmos_id shall be left NULL.
3. Call xp_target_extra_main with the target_id structure obtained in the previous step, select choice XP_TARG_EXTRA_MAIN_GEO, target_number shall be the num_user_target returned from the previous function and target_type XP_LOS_TARGET_TYPE, in order to obtain several output vectors (main_results, main_results_rate, main_results_rate_rate)
4. The main_results vector contains the geodetic latitude and altitude, as well as geodetic and geocentric longitude.
5. The geodetic latitude and altitude and the geocentric longitude shall be used in another call to xl_geod_to_cart function, in order to obtain the Earth Fixed Cartesian coordinates (X, Y Z) of that particular point.
6. Both Earth Fixed and geodetic/geocentric coordinates should be stored for later usage.

Once the EAF-FOV border has been projected into Earth Fixed coordinates, it must be used as a filter on all the points available in the DGG ADF.

As a first filter, the bounding box of the contour in lat-lon coordinates can be easily computed with the maximum and minimum values of latitude and longitude for all points in the contour. This guarantees that not all pixels from the DGG shall be passed to the second filter.

The second filter needs to be a bit more sophisticated, in order to take into account the strange geometry of the contour once it has been projected. This is done, for example, by taking the longitude of the pixel that needs verification of whether it is inside or outside the zone, and computing the intersections of this constant longitude with the contour. There can be several cases:

- No intersection: Means that the pixel is outside the zone
- Intersection in one point only: Means that the pixel outside the zone, or that one pole is inside the zone. If the pole is inside the zone, then if the latitude of the pixel is between the pole and the latitude of the intersection, the pixel is inside, otherwise it is outside
- Intersection on two points: If the latitude of the pixel is within the latitudes of the intersections, it is inside, otherwise it is outside

- Intersections on more than two points are expanded cases of the ones presented before.

Once the list of Earth Fixed pixels has been extracted, it is also interesting to use their latitude, longitude and altitude values to obtain the Earth Fixed Cartesian coordinates. This is done using the xl_geod_to_cart function described in previous points.

It is important to create at this point the distinction between land and sea pixels, according to the AUX_MASK ADF, which for each unique pixel id in the DGG contains a land and a sea flag for each id. It must be mentioned, that one pixel may have one or both flags at the same time (i.e. coastlines are to be processed as land and sea)

It should also be a good place to extract the RFI flag for the pixels as well, which will be later used in the flagging of measurements.

5.5.5. Error Handling

In case any error happens during projection of the EAF-FOV or extraction of the Earth Fixed pixels, the processing of the scene must be aborted. The other scenes can continue to be processed nominally.

5.6. Compute Pixel Angles

This computation shall be performed for each Earth Fixed pixel belonging to the scene, in order to compute all the relevant angles from S/C to pixel in several reference frames.

5.6.1. Inputs

- attitude_id
- earth_pixel (taken from earth_pixel_list)

5.6.2. Outputs

- satellite_to_target_sc_frame (azimuth and elevation)
- satellite_to_target_topo_frame (azimuth and elevation)
- target_to_satellite_topo_frame (azimuth and elevation)

5.6.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
attitude_id	EE CFI pointing structure	Section 5.2.3	xp_attitude_id	I	N/A	1

Variable name	Description	Definition	Type	Class	Unit	Size
earth_pixel	Pixel extracted from the list of pixels contained inside the EAF-FOV border	Section 5.5.3		I	N/A	1
satellite_to_target_sc_frame	S/C to pixel directions expressed in the S/C frame	N/A	double	O	deg	2
satellite_to_target_topo_frame	S/C to pixel directions expressed in topocentric frame	N/A	double	O	deg	2
target_to_satellite_topo_frame	Pixel to S/C directions expressed in topocentric frame	N/A	double	O	deg	2

Table 11: Compute Pixel Angles variable list

5.6.4. Implementation

Based on the pixel Earth Fixed Cartesian coordinates, a selection of EE CFI functions must be called in order to compute the respective angles. The procedure is the following:

1. Call xp_target_generic with attitude_id structure, the EF position of the pixel, in order to obtain an initialised target_id structure. Parameter dem_id shall be left NULL, deriv shall be XP_NO_DER
2. Call xp_target_extra_main with the target_id structure obtained in the previous step, select choice XP_TARG_EXTRA_MAIN_ALL, target_number shall be the num_user_target returned from the previous function and target_type XP_LOS_TARGET_TYPE, in order to obtain several output vectors (main_results, main_results_rate, main_results_rate_rate)
3. The results are gathered from the main_results vector in the following way:
   3.1. target_to_satellite_topo_frame[0] = main_results[8];/*azimuth target to S/C Topocentric CS*/
   3.2. target_to_satellite_topo_frame[1] = 90.0 - main_results[9];/*elevation target to S/C Topocentric CS, using complementary to set 0º on the pixel local vertical*/
   3.3. satellite_to_target_topo_frame[0] = main_results[4];/*azimuth S/C to target Topocentric CS*/
   3.4. satellite_to_target_topo_frame[1] = main_results[5];/*elevation S/C to target Topocentric CS*/
   3.5. satellite_to_target_sc_frame[0] = main_results[6];/*azimuth S/C to target Attitude CS*/
   3.6. satellite_to_target_sc_frame[1] = main_results[7];/*elevation S/C to target Attitude CS*/

5.6.5. Error Handling

In case of error during the angle computation, the specific pixel processing must be aborted. The rest of the pixels in the scene can continue to be computed with no adverse effects. There is no need to abort the scene processing, as only one pixel has been affected.

5.7. Apodisation Window Computation

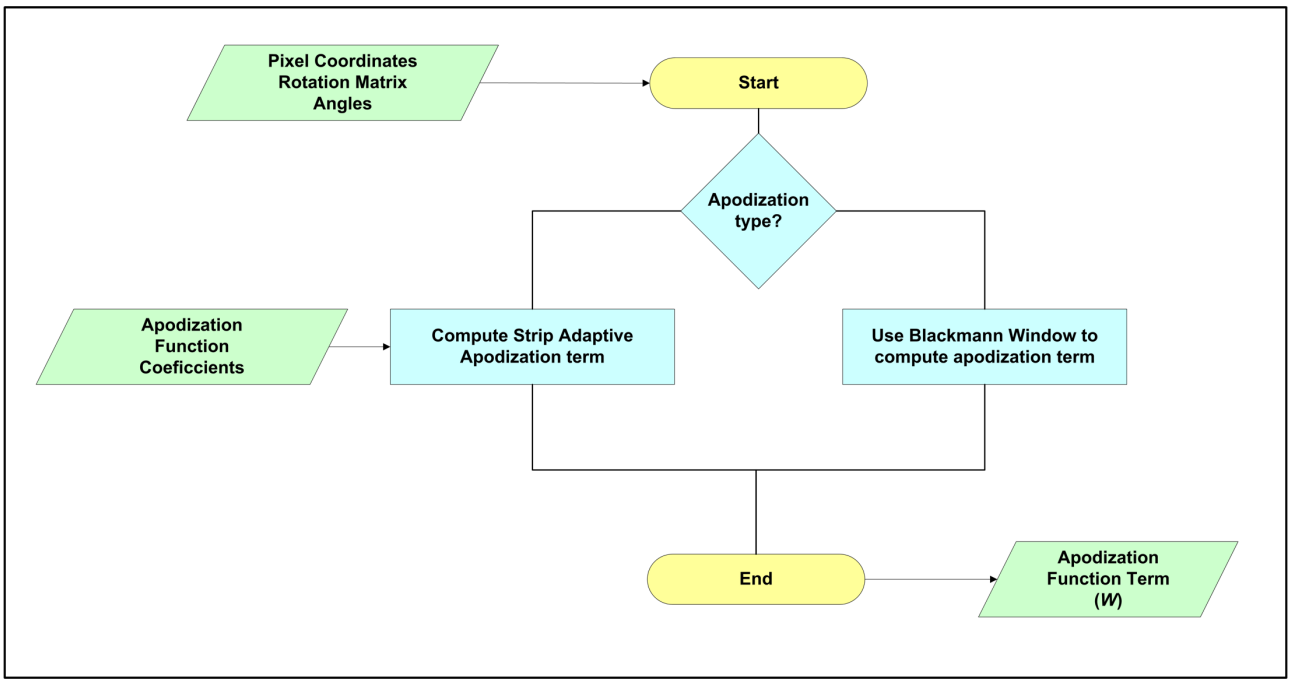

Figure 12: Apodisation Window module processing steps

The apodisation window computation will return the value of apodisation window for each pixel, based on the antenna frame coordinates. Optionally, it will compute the strip adaptive apodisation window term using also the (xi, eta) coordinates. The L1 Processor Prototype shall implement both apodisation methods, although it shall be considered that the baseline for the L1 Operational Processor shall not include Strip Adaptive processing.

5.7.1. Inputs

- Antenna plane coordinates (xi,eta), non-redundant baseline coordinates (u,v)
- attitude_id
- earth_pixel (taken from earth_pixel_list)
- Auxiliary Data
  - Apodisation function model coefficients (SM_xxxx_AUX_APDL__<ID>, _xxxx_AUX_APDS__<ID> or _xxxx_AUX_APDNRT<ID>)

5.7.2. Outputs

- Apodisation function term, W

5.7.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting

that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
xi_eta_pixel	Pixel xi-eta coordinates (antenna frame)	Eq. 13	double	I	N/A	2
attitude_id	EE CFI pointing structure	Section 5.2.3	xp_attitude_id	I	N/A	1
earth_pixel	Pixel extracted from the list of pixels contained inside the EAF-FOV border	Section 5.5.3		I	N/A	1
apod_model1_points	Pre-computed cubic coefficients for Strip Adaptive Apodisation model [$\log10(\alpha_u/\alpha_v)$]	Eq. 31	double	I	N/A	4
apod_model2_points	Pre-computed cubic coefficients for Strip Adaptive Apodisation model [$\log10(\alpha_u,\alpha_v)$]	Eq. 31	double	I	N/A	4
u_v	(u,v) coordinates of the instrument baselines	[RD.17]	t_l1c_baselines	I	N/A	NON_RED_BASELINES
xi_eta_circle	xi-eta coordinates of 6 points in a circle around the pixel projection (antenna frame)	Eq. 7	double	L	N/A	2x6
w_max	Major semi-axis of projected ellipse (antenna frame)	Eq. 31	double	L	N/A	1
w_min	Minor semi-axis of projected ellipse (antenna frame)	Eq. 31	double	L	N/A	1
delta	Angle between major and minor semi-axes	Eq. 28	double	L	N/A	1
alpha_u	Strip adaptive Apodisation coefficient	Eq. 26	double	L	N/A	1
alpha_v	Strip adaptive Apodisation coefficient	Eq. 26	double	L	N/A	1

Variable name	Description	Definition	Type	Class	Unit	Size
w	Apodisation term	Eq. 26	double	O	N/A	NON_RED_BASELINES

Table 12: Apodisation Window computation variable list

5.7.4. Implementation

The Blackman Window is used to compute the apodisation factor (Eq. 17):

```
//Compute w term using the Blackmann window
n_el=21;        //Nel for MIRAS instrument
d=0.875;        //Wavelengths between baselines
For(i=0... NON_RED_BASELINES)
   double term_a = PI*(sqrt(u_v[i]->u* u_v[i]->u + u_v[i]->v* u_v[i]-> v)) / (sqrt(3) *
                  n_el *d);
   w[i] = 0.42 + 0.5 * cos(term_a) + 0.08*cos(2*term_a);
EndFor
```

5.7.5. Error Handling

Error Nr	Error Description	Error Code Returned	Program behaviour	Quality flag raised
1	Error in interpolation for apodisation parameters	DATA_CORRUPTED	The element is set to a default value (i.e. Blackman default).	Yes

Table 13: Apodisation Window computation error handling

5.8. Add Artificial Scene and Compute Brightness Temperature

This computation shall be performed for each Earth Fixed pixel belonging to the scene, in order to perform a Discrete Fourier Transform. It shall perform a 2D integration, combining the xi-eta coordinates of the pixel, the Brightness Temperature Fourier components from the L1b product and the apodisation window computed before.

5.8.1. Inputs

- Direct interface (since merge of L1b and L1c processors in L1OP v710)

- Reconstructed $\hat{T}_B$ frequencies at top of the atmosphere
- Constant Earth BT subtracted if Gibbs 1 method was applied
- Frequencies of the artificial scene if Gibbs 2 method was applied

❑ satellite_to_target_sc_frame (azimuth and elevation)

❑ Apodisation function term, W

5.8.2. Outputs

❑ Pixel brightness temperature

5.8.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
scene_bt_frequencies_dual	Brightness Temperature Fourier components for the whole scene	Section 3.2.2	Double complex	I	N/A	1396
scene_bt_frequencies_full	Brightness Temperature Fourier components for the whole scene	Section 3.2.2	Double complex	I	N/A	2791
l1b_temps_freqs_data	Artificial Gibbs 2 scene in frequency domain	[AD.8]	double	I	N/A	11164
constant_earth_BT	Brightness Temperature for a constant Earth subtracted during L1b processing	Section 3.2.2	Double	I	K	1
satellite_to_target_sc_frame	S/C to pixel directions expressed in the S/C frame	Section 5.6.3	double	I	deg	2
w	Apodisation term	Eq. 26	double	I	N/A	1396
u_v	Non-redundant baselines coordinates	[RD.17]	t_l1c_baselines	I	N/A	1396
pixel_bt_measurement	Integrated Brightness Temperature for this scene at that particular viewing direction	N/A	Double complex	O	K	1

Table 14: Compute BT variable list

5.8.4. Implementation

First of all, it must be indicated that there are two processing approaches, depending if the L1b data is coming from dual polarisation or full polarisation. The only difference lies in the quantity of L1b data required, which is 1396 complex values for H or V scenes, and 2791 complex values for HV scenes.

The pixel xi-eta coordinates are extracted from the azimuth (ϕ) and elevation (θ) observation angles expressed in the S/C frame, according to equation 7:

The Brightness Temperature measurement for H and V scenes is computed by performing a Discrete Fourier Transform, equivalent to integration over each of the non-redundant baseline coordinates. The first baseline corresponds to the centre frequency, so it contains only a real value. The integration must be performed over the remaining 1395 baselines, taking into account that the data available in each L1b scene is only half of the star domain.

As it has been described in [AD.8], each Brightness Temperature Frequency component corresponds to a specific *u,v* baseline. The integration is performed over one half of the star domain by multiplying the 1395 elements in the BT frequencies vector element by element with the apodisation window vector and with an equivalent vector containing the exponential term. Then it must be completed with the other half of the star domain, by simply changing the sign of the *u,v* baseline and doing the complex conjugate of the BT frequencies:

$$T(\xi,\eta)=\frac{\sqrt{3}d^2}{2}\sum_{u,v}\hat{T}(u,v)\cdot W(u,v)\cdot e^{2\pi i(u\xi+v\eta)}+\frac{\sqrt{3}d^2}{2}\sum_{-u,-v}\hat{T}^*(u,v)\cdot W(u,v)\cdot e^{2\pi i(u\xi+v\eta)} \qquad \textit{Eq. 35}$$

The apodisation window should always be symmetrical, so the same term can be used for the *u,v* baseline or for the *-u,-v* baseline.

After computing the Brightness Temperature from the frequency components, the extended sources that may have been subtracted during L1b processing, in order to perform Gibbs error mitigation, have to be added back. Depending on the Gibbs level it can be a constant flat Earth (Gibbs=1), or a constant flat Land plus a model Ocean (Gibbs=2). The flat Earth or Land values are annotated in the L1b product, whereas the Ocean model is the same as described in section 2.2.2.1.4 of [AD.8]. The Brightness Temperatures are added after a DFT is applied to the L1b Gibbs 2 Visibilities, obtained with the same apodization as the measurements.

This is done by applying a DFT to the artificial scene frequencies from the BTs extended from 196x196 to 256x256 estimated in L1b. The DFT is applied t with a Blackmann Window to obtain the apodized artificial BTs and the results are summed pixel-by-pixel to the delta BTs from Eq. 35.

In full polarisation measurement, the H and V scenes are processed in the same way as above, whereas the HV scenes are processed by combining the 2791 complex values available. In this case, there is no need to compute the complex conjugate for the *-u,-v* baselines, as described in [AD.8], the HV BT frequencies cover the whole star domain so the DFT can be performed in a straightforward manner.

In full polarisation scenes (HV pol) there is no Gibbs term to be added back, it is only applicable to dual polarisation scenes.

5.8.5. Error Handling

No errors are expected, unless wrong numbers (i.e. NaN) are coming from previous computations or from the L1b products. The type of computation required is done inside a for loop, with no possibility of division by zero or any other mathematical error.

5.9. Compute Radiometric Accuracy

This computation shall be performed for each Earth Fixed pixel belonging to the scene, in order to extract an estimation of the Radiometric Accuracy for the Brightness Temperature computed before. The accuracy for each measurement will vary according to the xi-eta coordinates it has.

Several auxiliary values are required here, based on equation 9.

5.9.1. Inputs

- satellite_to_target_sc_frame (azimuth and elevation)
- W is the apodisation window term for each (u,v) baseline
- Auxiliary Data
  - Average Antenna Patterns (SM_xxxx_AUX_PATT<ID>)
    - $G^{pq}(\xi,\eta)$ is the averaged LICEF receiver directional power Gain function normalised so that it is unity at boresight.
  - Default PLM values (SM_xxxx_AUX_PLM___<ID>)
    - f_0 is the central frequency and f_{01} is the low frequency
    - D is the averaged directivity of the LICEFs for each polarisation.
- L1a HKTM Data (SM_xxxx_TLM_MIRA1A_<ID>)
  - T_{sys}^{pq} is the averaged System Temperature measured by the PMS system, which has been used to de-normalise the L1a calibrated visibilities.
- d is the distance ratio between receivers (0.875)
- B is the equivalent receiving frequency bandwidth in Hz, currently being 19 MHz
- τ_{eff} is the effective integration time, and is equivalent to τ / c_{eff} where τ is the integration time (1.2s for H and V scenes and 0.8s for HV scenes) and c_{eff} is the coefficient that accounts for the 1-bit correlation, oversampling and hermiticity (c_{eff}=1.81).

5.9.2. Outputs

- pixel_radiometric_accuracy

5.9.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
satellite_to_target_sc_frame	S/C to pixel directions expressed in the S/C frame	Section 5.6.3	double	I	deg	2
w	Apodisation term	Eq. 26	double	I	N/A	1396
D	Antenna directivity	Section 3.2.3	double	I	N/A	1
Center_frequency	Central frequency of operation of the instrument	Section 3.2.3	double	I	MHz	1
Low_frequency	Lower limit frequency of operation of the instrument	Section 3.2.3	double	I	MHz	1
B	Frequency Bandwidth	Section 3.2.3	double	I	MHz	1
Avg_Tsys	System Temperatures average	Section 3.2.3	double	I	K	1
Avg_Pattern	Antenna Pattern average module	Section 3.2.3	double	I	N/A	181x181
R	the redundancy level of each (u,v) baseline (i.e. number of times that the baseline has been measured, 1 for non-redundant baselines, greater than 1 for the rest)	Section 3.2.3	double	L	N/A	1396
pixel_radiometric_accuracy		N/A	Double	O	K	1

Table 15: Compute Radiometric Accuracy variable list

5.9.4. Implementation

As explained initially in section 3.2.3, the following equation shall be used to compute the radiometric accuracy for a given viewing direction:

$$\Delta T^{pq}(\xi,\eta) = \frac{\Omega^{pq}\sqrt{1-\xi^2-\eta^2}}{\bar{G}^{pq}(\xi,\eta)}\frac{\sqrt{3}}{2}d^2\frac{T_{sys}^{pq}}{\sqrt{B\cdot\tau_{eff}}}\alpha_w\alpha_{ol} \qquad \textit{Eq. 36}$$

Where the different parameters are computed as follows:

- The pixel xi-eta coordinates are extracted from the azimuth (ϕ) and elevation (θ) observation angles expressed in the S/C frame, according to equation 7:
- Ω^{pq} is the solid angle of the antenna, and can be computed as $\frac{4\pi}{D}$ for the current LICEFs, where D is the averaged directivity of the LICEFs. D is taken from the AUX_PLM_ ADF file for each polarisation. For full polarisation, D^{HV} is computed as $\sqrt{D_H D_V}$.
- $\bar{G}^{pq}(\xi,\eta)$ is the averaged LICEF receiver directional power gain function normalised so that it is unity at boresight. This data shall be retrieved from the average antenna pattern ADF (AUX_PATT__) for each polarisation, and the value corresponding to the exact xi-eta coordinates extracted by interpolation. For full polarisation, $\bar{G}^{HV}(\xi,\eta)$ is computed as $\sqrt{G^{HH}G^{VV}}$.
- d is the distance ratio between receivers (0.875)
- T_{sys}^{pq} is the averaged System Temperature measured by the PMS system, which has been used to de-normalise the L1a calibrated visibilities. These values are taken from the corresponding L1a HKTM product, for the ancillary data produced at the same OBET time as the scene being processed.
- B is the equivalent receiving frequency bandwidth in Hz, currently being 19 MHz. B is taken from the AUX_PLM_ ADF file.
- τ_{eff} is the effective integration time, and is equivalent to τ/c_{eff} where τ is the integration time and c_{eff} is the coefficient that accounts for the 1-bit correlation, oversampling and hermiticity (c_{eff}=1.81). For pure scenes (H and V pol) the integration time is 1.2s, for mixed scenes is 0.46s (H and V pol), and for HV real and imaginary it is 0.64s (i.e. T3 and T4).
- $\alpha_w = \sqrt{\sum_u\sum_v \frac{(W(u,v))^2}{R(u,v)}}$ accounts for the apodisation window and redundancies in the measurements, where W is the apodisation window term for each (u,v) baseline and R is the redundancy level of that same baseline (i.e. number of times that the baseline has been measured, 1 for non-redundant baselines, greater than 1 for the rest)
- $\alpha_{ol} = \sqrt{1+e^{-2\pi\left(\frac{f_0-f_{01}}{B}\right)^2}}$ accounts for the local oscillator factor, where B is the bandwidth mentioned before, f_0 is the central frequency and f_{01} is the low frequency. Both frequency values f_0 and f_{01} are taken from the AUX_PLM_ ADF file

In order to compute the redundancy level R, the following computations must be performed, combining real time data with auxiliary data:

- First the LICEF failures (if any) must be retrieved from the L1a HKTM data whose OBET corresponds to the OBET of the scene being processed. The default failures stored in the AUX_FAIL__ ADF shall also be considered.
- Secondly, the LICEF coordinates of all LICEFs shall be extracted from the AUX_PLM__ ADF, together with the central frequency value f_0
- For each LICEF-LICEF pair, and in case they either one or the other are not failing, compute the u,v baseline coordinates using the LICEF positions and central frequency. This will point to one unique position in the star domain u_v vector, which should be incremented by one to reflect that the baseline has been measured.

After running through all possible LICEF pairs, the output shall be a vector R, which for each u_v position in the star domain will contain the number of times that the baseline was measured by the instrument in that scene.

For the 0,0 baseline, the number of redundancies shall be the number of NIR measurements available for that scene (3 if no failures, less in other cases).

5.9.5. Error Handling

In case applicable ADF files cannot be found, the processing must be aborted. The same case applies if the average System Temperatures cannot be obtained for the corresponding OBET.

If the LICEF Failures cannot be obtained for the corresponding OBET, the default values can always be used, but it is most probably that if the LICEF failures cannot be obtained for an OBET, most probably the System Temperatures will not be obtained either, reverting to the initial case.

In case the LICEF failures are too many, it may be possible that some of the values in the Redundancy vector are computed as zero. In this case, it is not advisable to use it in the equation, as it will cause a division by zero exception. A warning should be issued, and a default value of one should be used, in order to get a more reasonable value of the redundancy.

5.10. Compute Pixel Rotations

This computation shall be performed for each Earth Fixed pixel belonging to the scene, in order to extract the ionospheric and geometric polarisation rotation angles. These values are dependant on the viewing angles, that is why they must be computed on a pixel basis.

Previously computed values are required here, as it has been shown in sections 3.1.3 and 3.1.4.

It should be noted that the implementation description has been expanded to include both methods, although the baseline is only the Duesmann & Zundo implementation. The Waldteufel & Caudal implementation is maintained for historical reasons.

5.10.1. Inputs

5.10.1.1. Waldteufel and Caudal Implementation

- satellite_to_target_sc_frame (azimuth and elevation)
- satellite_to_target_topo_frame (azimuth and elevation)
- tilt angle
- local north angle
- Total Electron Content Data (TEC)
- Magnetic Field Strength (F)
- Magnetic Field inclination (I) and declination (D)

5.10.1.2. Duesmann and Zundo Implementation

- satellite_to_target_sc_frame (azimuth and elevation)
- satellite_to_target_topo_frame (azimuth and elevation)
- satellite position in EF
- rotation matrix between BFP and EF
- Total Electron Content Data (TEC)
- Magnetic Field Strength (F)
- Magnetic Field inclination (I) and declination (D)

5.10.2. Outputs

- Ionospheric_rotation_angle
- Geometric_rotation_angle

5.10.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
satellite_to_target_sc_frame	S/C to pixel directions expressed in the S/C frame	Section 5.6.3	Double	I	deg	2
satellite_to_target_topo_frame	S/C to pixel directions expressed in topocentric frame	Section 5.6.3	Double	I	deg	2

Variable name	Description	Definition	Type	Class	Unit	Size
geomag_f	Magnetic field strength	Section 3.1.1	double	I	Tesla	1
geomag_d	Magnetic field declination	Section 3.1.1	double	I	degrees	1
geomag_i	Magnetic field inclination	Section 3.1.1	double	I	degrees	1
tec	Total Electron Content	Section 3.1.1	double	I	TEC	1
tilt_angle	Angle between S/C to Nadir direction and Instrument boresight	Section 5.4	double	I	deg	1
local_north_angle	Angle between S/C ground track instantaneous direction and local north	Section 5.4	double	I	deg	1
satellite_position_EF	Position of the satellite in the EF frame	N/A	double	I	N/A	3
matrix_BFP2EF	Rotation matrix between the BFP and the EF frame.	Section 5.2	double	I	N/A	9
Ionospheric_rotation_angle	Faraday rotation angle	Section 3.1.3	double	O	degrees	1
Geometric_rotation_angle	Rotation angle due to geometry	Section 3.1.4	double	O	degrees	1

Table 16: Compute Pixel Rotations variable list

5.10.4. Implementation

The Faraday rotation angle is easily computed once the TEC, geomagnetic field components and pixel angles are available. Using equation 1, the following advice should be heeded:

- All angles should be transformed to radians before using them inside the equation
- TEC should be expressed in TECU
- Geomagnetic field should be expressed in Tessla
- θ_g ϕ_g are the satellite_to_target_topo_frame angles

$$\omega = -6950 \times F \times TEC \times \left[\sin I + \cos I \times \tan \theta_g \times \cos\left(\phi_g - D\right) \right] \qquad \textit{Eq. 37}$$

It must be remarked that the value of θ_g above is not computed directly as the elevation in satellite_to_target_topo_frame, as the EE CFI functions will return a negative angle. It has to be computed as shown in Eq.40.

Regarding ϕ_g, to be applied in Eq.37 it must be computed as obtained directly from the EE CFI, i.e. satellite_to_target_topo_frame[0]. This is because in Eq.37 ϕ_g must be oriented in the same axis as D, which is consistent with the EE CFI azimuth axis.

5.10.4.1. Waldteufel and Caudal Implementation

For the geometric rotation angle, an initial computation must be performed, in order to obtain the real tilt angle of the array, as well as the angle of the sub-satellite track with the local north. These should have been computed as indicated in section 5.4.

The equations shown in section 3.1.4 can be computed as follows:

$$\theta = \frac{\pi}{2} - \text{satellite\_to\_target\_sc\_frame}[1] \qquad \textit{Eq. 38}$$

$$\phi = -\frac{\pi}{2} + \text{satellite\_to\_target\_sc\_frame}[0] \qquad \textit{Eq. 39}$$

$$\theta_g = \frac{\pi}{2} + \text{satellite\_to\_target\_topo\_frame}[1] \qquad \textit{Eq. 40}$$

$$\phi_g = \frac{\pi}{2} - \text{satellite\_to\_target\_topo\_frame}[0] + \text{local\_north} \qquad \textit{Eq. 41}$$

Depending on the magnitude of ϕ_g, the value of φ must be computed differently:

$$\varphi = \arcsin\left(\frac{\cos(tilt\_angle)\sin\theta_g - \sin(tilt\_angle)\cos\theta_g\sin\phi_g}{\sin\theta}\right)\left\{\frac{\pi}{2} \le \phi_g \le \frac{3\pi}{2}\right.$$
$$\varphi = \pi - \arcsin\left(\frac{\cos(tilt\_angle)\sin\theta_g - \sin(tilt\_angle)\cos\theta_g\sin\phi_g}{\sin\theta}\right)\left\{otherwise\right. \qquad \textit{Eq. 42}$$

The final geometric rotation angle must be computed as **Geometric_rotation_angle $\alpha = \phi - \varphi$**

5.10.4.2. Duesmann and Zundo Implementation

Setting

$$\theta_e = \text{target\_to\_satellite\_topo\_frame}[1] \qquad \textit{Eq. 43}$$

$$\varphi_e = \text{target\_to\_satellite\_topo\_frame}[0] \qquad \textit{Eq. 44}$$

in equations 7 and 8, the emission vectors are calculated in the topocentric frame.

With the CFI function XL_EF_TO_TOPOCENTRIC the rotation matrix from the BFP to EF is transformed to the rotation matrix from the BFP to the pixel topocentric frame. Note that at this point, the rotation matrix is written in CFI notation and it must be converted to SMOS notation, i.e.,

$$\begin{cases} X_{CFI} = -X_{MIRAS} \\ Y_{CFI} = -Y_{MIRAS} \\ Z_{CFI} = Z_{MIRAS} \end{cases} \quad \textit{Eq. 45}$$

This matrix allows the conversion of the emissions vectors in topocentric coordinates to MIRAS coordinates.

With

$$\theta_s = \text{satellite\_to\_target\_sc\_frame}[1] \quad \textit{Eq. 46}$$

$$\varphi_s = \text{satellite\_to\_target\_sc\_frame}[0] \quad \textit{Eq. 47}$$

being used to determine the vector from satellite to target (equation 9) the auxiliary variables are computed and the Ludwig vectors computation is performed.

The geometric rotation angle is then taken from

$$\alpha' = \arctan\left(\frac{\vec{Eh}_{MIRAS} \cdot \vec{L}_{y}^{\,MIRAS}}{\vec{Eh}_{MIRAS} \cdot \vec{L}_{x}^{\,MIRAS}} \right) \quad \textit{Eq. 48}$$

5.10.5. Error Handling

In equation 42, depending on the values inside the arcsin function, there can be a NaN result if the numerator is bigger than the denominator. In order to rule out limit cases within the tolerances, if the ratio is slightly higher than 1, the value for φ shall be π/2, whereas if it is slightly smaller than –1, φ shall be –π/2.

All other cases shall be treated as errors, and the computation of the affected pixel shall be skipped.

5.11. Compute Pixel Footprint

This computation shall be performed for each Earth Fixed pixel belonging to the scene, in order to extract the footprint semi-major and semi-minor axes, as it is approximated by an ellipse. The synthetic beam shape of the antenna intersecting the antenna frame forms the footprint, once it is projected over the Earth's surface.

The common agreement is that the footprint is defined by the –3dB contour of the Equivalent Array factor described in equation 10.

In order to avoid computing the Array Factor and projecting it for a lot of points and then looking for the maximum and minimum values, it can be established that the maximum elongation of the ellipse is achieved for the direction formed by the S/C nadir point and the pixel, whereas the minimum elongation shall be achieved in a direction orthogonal to that.

5.11.1. Inputs

5.11.1.1. Ellipse Fit Method

- attitude_id
- earth_pixel (taken from earth_pixel_list)
- satellite_to_target_sc_frame (azimuth and elevation)
- satellite_to_target_topo_frame (azimuth and elevation)
- local north angle
- W is the apodisation window term for each (u,v) baseline
- Auxiliary Data
  - Default PLM values (SM_xxxx_AUX_PLM___<ID>)
    - f_0 is the central frequency

5.11.1.2. Analytical Pixel Footprint

- attitude_id
- earth_pixel (taken from earth_pixel_list)
- satellite_to_target_sc_frame (azimuth and elevation)
- satellite_to_target_topo_frame (azimuth and elevation)
- matrix_BFP2EF
- satellite_position_EF

5.11.2. Outputs

- Footprint_maximum
- Footprint_minimum

5.11.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
attitude_id	EE CFI pointing structure	Section 5.2.3	xp_attitude_id	I	N/A	1

Variable name	Description	Definition	Type	Class	Unit	Size
earth_pixel	Pixel extracted from the list of pixels contained inside the EAF-FOV border	Section 5.5.3		I	N/A	1
satellite_to_target_sc_frame	S/C to pixel directions expressed in the S/C frame	Section 5.6.3	double	I	deg	2
satellite_to_target_topo_frame	S/C to pixel directions expressed in topocentric frame	Section 5.6.3	double	I	deg	2
local_north_angle	Angle between S/C ground track instantaneous direction and local north	Section 5.4	double	I	deg	1
w	Apodisation term	Eq. 26	double	I	N/A	1396
B	Frequency Bandwidth (19MHz)	Section 3.3.2	double	I	MHz	1
Center_frequency	Central frequency of operation of the instrument	Section 3.3.2	double	I	MHz	1
u_v	Set of non-redundant baselines (0 – redundant, 1 – non-redundant)	[RD.17]	t_l1c_baselines	I	N/A	1396
matrix_BFP2EF	Rotation matrix between the BFP and the EF frame.	Section 5.2	double	I	N/A	9
satellite_position_EF	Position of the satellite in the EF frame	N/A	double	I	N/A	3
Footprint_maximum	Maximum size of projected elliptical footprint	N/A	double	O	km	1
Footprint_minimum	Minimum size of projected elliptical footprint	N/A	double	O	km	1

Table 17: Compute Pixel Footprint variable list

5.11.4. Implementation

There are two selectable methods that can be used to compute the footprint axes. The first one is the original method that requires the projection of a minimum of 6 points of the footprint in order to be fit into an ellipse. The second one computes the analytical expression of the footprint doing certain approximations, and uses the analytical formula to derive the footprint axes.

5.11.4.1. Ellipse Fit Method

There are two main computations to be performed here. The first one shall be used to identify the xi-eta coordinates of the two points in the –3dB contour of the Array Factor, one along the S/C to pixel azimuth direction, and the other in the orthogonal one.

The second step shall project those points over the Earth, in order to compute the geodetic distance between the pixel centre and them.

The first step must be done by approximating the footprint shape on the ground by an ellipse. For this fit, 6 projected points are the minimum required.

Initially, we must use equation 10 for the computation of the Array Factor:

- The pixel xi-eta coordinates are extracted from the azimuth (ϕ) and elevation (θ) observation angles expressed in the S/C frame, according to equation 7.
- The six azimuth directions over which the AF needs to be computed can be spaced 60º in order to provide maximum coverage:

$$\phi_F = i \times 60^{\circ} \quad (0 \le i \le 5) \qquad \textit{Eq. 49}$$

- And the equation that needs to be solved is the following one, in order to find Δr for each of the previous azimuth angles:

$$\frac{\sum_m \sum_n W(u_{mn}, v_{mn})}{2} = \sum_m \sum_n W(u_{mn}, v_{mn}) \cdot \rho\left(-\frac{u_{mn} \cdot \xi + v_{mn} \cdot \eta}{f_0}\right) \cdot e^{j2\pi\left(u_{mn}\Delta r \sin\phi_p + v_{mn}\Delta r \cos\phi_p\right)} \qquad \textit{Eq. 50}$$

- The FWF function can be approximated by $\rho(\tau) \approx e^{\left[-\pi W_r^2 (u\xi + v\eta)^2\right]}$, where W_r is computed as B/f_0, and both of these values are taken from the AUX_PLM ADF. The calibrated FWF shape product could also be used here, although the complexity increase for the gain in the accuracy computation does not make it worthwhile.
- The equation is easily solved by using the secant method, using the maximum at Δr=0, and as start value of Δr half of the Blackman ideal –3dB contour. This Blackman ideal –3dB contour value is computed as:

$$\Delta r_{Blackman-3dB} = \frac{2.321919}{\left(2\sqrt{3} \cdot NEL \cdot d\right)^{0.972282}} \qquad \textit{Eq. 51}$$

- In the ideal case, and if the Blackman window has been used as apodisation window, the value presented before can be used as Δr, in order to compute the xi-eta coordinates of the six looked-for values

$$\left.\begin{aligned}\xi_F &= \xi + \Delta r_F \sin\phi_F \\ \eta_F &= \eta + \Delta r_F \cos\phi_F\end{aligned}\right\} \qquad \textit{Eq. 52}$$

Once the values have been found, the projection on Earth is done using functions already explained before:

1. For each xi-eta point extract the corresponding azimuth and elevation angles. Azimuth must be converted to [0,360] range before being used by CFI functions.
2. Call xp_target_inter function with attitude_id structure, the azimuth and elevation angles, deriv XP_NO_DER and inter_flag XP_INTER_1ST, in order to obtain an initialised target_id structure. Parameter dem_id and atmos_id shall be left NULL.
3. Call xp_target_extra_main with the target_id structure obtained in the previous step, select choice XP_TARG_EXTRA_MAIN_GEO, target_number shall be the num_user_target returned from the previous function and target_type XP_LOS_TARGET_TYPE, in order to obtain several output vectors (main_results, main_results_rate, main_results_rate_rate)
4. The main_results vector contains the geodetic latitude and altitude, as well as geodetic and geocentric longitude.
5. The geodetic latitude and the geocentric longitude shall be used in another call to xl_geod_distance function together with the latitude and longitude of the pixel centre, in order to retrieve the geodetic distance and the relative azimuth from the centre to each point.

Once the distance and relative azimuths for each of the 6 points is known, they can be used in any least squares method to fit the 6 parameters of an ellipse quadratic equation, for example [RA.02].

Once the ellipse equation is known, obtaining both semi-axes is trivial.

5.11.4.2. Analytical Footprint Method

This method computes the analytical equation of the 3dB contour over the Earth's surface, and through that equation (approximated by an ellipse) the semi-axis and orientation can be computed.

First the equation of the aligned cone in the red reference frame of Figure 6 must be obtained. For that purpose, two rotations are performed, first the axis' cone is aligned with the $z_1 0 y_1$ plane and then a second rotation is done around the x_1 axis, with an angle of $(\theta - 90^o)$.

This creates a referential where the cone has the following equation:

$$x_2^2 + \cos^2(\theta) y_2^2 - r_{-3dB}^2 z_2^2 = 0 \qquad \textit{Eq. 53}$$

Using the rotation matrix between the BFP and the EF frame and with the CFI function XL_EF_TO_TOPOCENTRIC this equation is transformed to the topocentric frame of the target as a conic section. From it the major and minor semi axes of the ellipse are derived.

For further details on the implementation of this algorithm, please refer to [RD.13].

5.11.5. Error Handling

In case any error is returned by the CFI functions, the processing of this pixel shall be stopped, although the processing for the remaining pixels in the scene can continue.

5.12. Compute Pixel Flags

This computation shall be performed for each Earth Fixed pixel belonging to the scene, in order to flag it for different position within the FOV, or for corrections that were performed along the processing.

As a starting point, the pixel will inherit the L1b flags of the whole scene. These include the polarisation mode, and the corrections for direct Sun, direct Moon and reflected Sun.

Afterwards, the pixel shall be flagged according to its position in the FOV (if it is inside the strict AF-FOV or not, or if it is very close to the border of the Earth aliases, or to the Unit Circle aliases (also known as the "suspenders and belt"). Information about the RFI contamination shall be taken from the AUX_RFI ADF and from the L1b flags.

Finally, if the pixel is located very close to a Sun or Moon alias, it shall be flagged as well, and also if it is potentially affected by Sunglint.

Since L1OP v710, an extra set of flags is raised if the pixel is in the expected tails of an RFI source. The system response to each RFI source and the full expected tail contamination mask is computed in L1b processing (please see the DPM L1b, [AD.8], Section 2.2.2.1.8.) and used in this module to flag the pixel according to the expected impact.

5.12.1. Inputs

- Direct interface (since merge of L1b and L1c processors in L1OP v710)
  - l1b_flags
  - position of Sun and Moon in xi-eta coordinates
  - RFI System response frequencies
- Auxiliary Data
  - RFI List of sources with Latitude, Longitude, BT and action to take in the flagging routines (SM_xxxx_AUX_RFILST_<ID>).
  - Bi-static scattering coefficients (SM_xxxx_AUX_BSCAT_<ID>)
- earth_pixel (taken from earth_pixel_list)
- satellite_to_target_sc_frame (azimuth and elevation)
- earth_eaf_fov_list (projected EAF-FOV, as computed in section 5.5)

5.12.2. Outputs

- pixel_l1c_flags

5.12.3. List of variables

The following table describes the variables used in the subsequent implementation section. Variables are listed as input, local and output (I, L, O). The *Size* column indicates the number of elements constituting that variable, and NOT the size of the variable in bytes (this information can be taken from the *Type* column).

Variable name	Description	Definition	Type	Class	Unit	Size
earth_pixel	Pixel extracted from the list of pixels contained inside the EAF-FOV border	Section 5.5.3		I	N/A	1
satellite_to_target_sc_frame	S/C to pixel directions expressed in the S/C frame	Section 5.6.3	double	I	deg	2
l1b_flags	Scene flags taken from the L1b scene being processed	N/A	bit	I	N/A	8
l1b_rfi_freqs_data	RFI System Response in frequency domain	[AD.8]	double	I	N/A	2791
Sun_position	Xi-eta position of the Sun in the antenna frame	N/A	double	I	N/A	2
Moon_position	Xi-eta position of the Moon in the antenna frame	N/A	double	I	N/A	2
Earth_eaf_fov_list	List of lat-lon points in the border of the projected EAF-FOV	Section 5.4.3		I	N/A	1
rfi_pixel_list	Geographic coordinates for RFIs listed in AUX_RFILST		double	I	N/A	[Nbr_RFI_in_Snapshot]
pixel_l1c_flags	Pixel flags computed	N/A	bit	O	N/A	16

Table 18: Compute Pixel Flags variable list

5.12.4. Implementation

The computation of the pixel_l1c_flags should begin with the simple transfer of the l1b_flags of the scene being processed.

Another simple task is to extract from the RFI ADF the flag relative to the pixel being processed, using the pixel unique identifier.

All pixels being processed here will always be by definition inside the EAF-FOV. In order to check id the pixel is inside the strict AF-FOV, the following steps must be followed:

- The pixel xi-eta coordinates are extracted from the azimuth (ϕ) and elevation (θ) observation angles expressed in the S/C frame, according to equation 7.
- For all the image replica centres, as described in equation 28, the distance between all centres and the xi-eta point shall be computed
- If all distances are greater than 1, then the pixel is inside the strict AF-FOV. If any of the 6 computed distances is smaller than 1, then the pixel is outside the strict AF-FOV

In order to check if the pixel is close to the border, a configurable distance value must be used (it shall be defined in the L1 algorithm configuration file by the FOV_Border_Flag_Size flag). The computation must be performed using the pixel xi-eta coordinates, and projected EAF-FOV contour xi-eta coordinates.

The geometric distance between the pixel and each line formed by two consecutive points in the EAF-FOV contour shall be computed, using the formulation:

$$d = \sqrt{(\xi_{EAF} - \xi_{pix})^2 - (\eta_{EAF} - \eta_{pix})^2} \qquad \textbf{Eq. 54}$$

The condition that must be fulfilled to guarantee that the pixel is closer than 0.01 to the pixel (configurable) is:

- The computed distance d is smaller than 0.01

In addition, in order to check if the pixel is close to the Unit Circle Border aliases (suspenders and belt) the following procedure will be taken:

- The xi-eta coordinates for the 3rd to 5th unit circle replicas are computed, according with the formulation:

$$\alpha_{replica\_i} = \frac{\pi}{6} + i\frac{\pi}{3}, \text{ with i =3,4,5}$$

$$\xi_{replica\_i} = \frac{2\cos(\alpha_{replica\_i})}{\sqrt{3DisRat}} \quad \text{with i =3,4,5} \qquad \textbf{Eq. 54 a)}$$

$$\eta_{replica\_i} = \frac{2\sin(\alpha_{replica\_i})}{\sqrt{3DisRat}}$$

- The distance between the replicas border and the pixel is computed:

$$d_{replica\_i} = \sqrt{(\xi_{replica\_i} - \xi_{pix})^2 - (\eta_{replica\_i} - \eta_{pix})^2} \text{ with i =3,4,5} \qquad \textbf{Eq. 54 b)}$$

- The conditions that must be fulfilled to guarantee that the pixel is closer than 0.01 of one of the suspenders or belt are:

$$\begin{aligned} d_{replica\_i} &> 1 - \text{FOV\_Border\_Flag\_Size flag} \\ d_{replica\_i} &< 1 + \text{FOV\_Border\_Flag\_Size flag} \end{aligned} \quad \text{with i =3,4,5} \qquad \textit{Eq. 54 c)}$$

Therefore the width of the suspenders and belt is twice the FOV_Border_Flag_Size flag while the width of the border is FOV_Border_Flag_Size flag .

In order to compute the flag due to Sun or Moon alias being too close to the pixel, the following steps must be followed:

- The pixel xi-eta coordinates are extracted from the azimuth (ϕ) and elevation (θ) observation angles expressed in the S/C frame, according to equation 7.
- The Sun or Moon xi.eta coordinates are extracted from the L1b scene being processed. These coordinates represent the real xi-eta coordinates in the unit circle.
- For all the image replica centres, as described in equation 28, the distance between the pixel and the point located at replica centre plus Sun/Moon xi-eta coordinates shall be computed. This will compute the xi-eta coordinates of the aliases that fall inside the unit circle, but come from an aliased image.
- If the distance is smaller than a given threshold (0.075/0.01 for the Sun/Moon, computed considering a 1º beam around the Sun centre), then the pixel shall be flagged with the Sun/Moon alias flag

The Sun alias threshold is configurable using the Sun_Point_Flag_Size of the CNFL1P configuration file.

For the flagging of the pixels that could be affected by the Sun tails, the following image shall be taken into consideration. This figure shows the expected propagation path of the tails due to the Hexagonal Discrete Fourier Transform applied to L1b data:

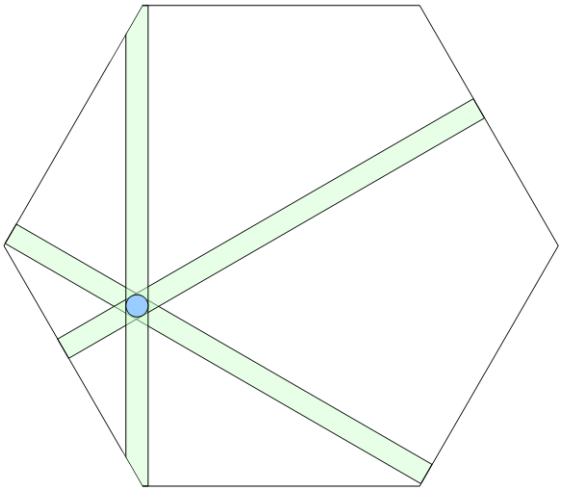

Figure 13: Sun tails areas

The blue circle represents the aliased Sun entering in the nominal reconstruction hexagon. The green areas represent the expected paths of ripple propagation due to a strong source in the blue circle. The radius of the blue circle and the width of the green areas are the same as presented before (0.01 units in xi-eta, equivalent to a 1º beam diameter). The method to compute the Sun tails flag would then be the following:

- The pixel xi-eta coordinates are extracted from the azimuth (ϕ) and elevation (θ) observation angles expressed in the S/C frame, according to equation 7.
- The Sun xi.eta coordinates are extracted from the L1b scene being processed. These coordinates represent the real xi-eta coordinates in the unit circle.
- For all the image replica centres, only one Sun alias will enter into the nominal hexagon area. Its coordinates $(\xi_{SunAlias}, \eta_{SunAlias})$, shall be computed after identifying which of the aliases falls inside the hexagon.
- The distances from the pixel xi-eta coordinates to the three lines parting from the center of the Sun alias shall be computed, using the following equations

$$d_1 = \left|\xi - \xi_{SunAlias}\right|$$
$$d_2 = \frac{\left|\left(\xi - \xi_{SunAlias}\right) + \sqrt{3}\left(\eta - \eta_{SunAlias}\right)\right|}{2}$$
$$d_3 = \frac{\left|\left(\xi - \xi_{SunAlias}\right) - \sqrt{3}\left(\eta - \eta_{SunAlias}\right)\right|}{2}$$

Eq. 55

- If any of the distances is smaller than a given threshold (0.01, computed considering a 1º beam around the Sun centre), then the pixel shall be flagged with the Sun tails flag

For what concerns the flagging of pixels if they are potentially affected by Sunglint, the algorithm to compute this flag is described in the following steps:

- In order to detect potential Sunglint contamination, the modelled Sunglint computed in L1b processing shall be compared against a configurable threshold. This flag must be computed in L1c due to the inherent differences in the working pixel grids between L1b (128x128 unit circle) and L1c (4H9 ISEA), which provide a different pixel geometry
- The equation that must be implemented is almost identical to Eq.34 of DPM L1b, but with the following modification:

$$\bar{T}^{pq}_{B\,sunglint}(\xi,\eta) = \frac{\sigma^0_{pp}\left(\vec{n}_s,\vec{n}_i\right)+\sigma^0_{pq}\left(\vec{n}_s,\vec{n}_i\right)}{4\pi\cos\theta} \qquad \textit{Eq. 56}$$

- We are not interested in the Brightness Temperature value per se, but only in the geometric relationship between S/C–pixel–Sun, so we take out the dependency on the Sun BT and solid angle. This way the flagging depends solely on the predicted scattering and not on the state of the Sun.
- The equation above needs to take into consideration all of the points reflected in section 2.2.2.1.6 of DPM L1b, namely in the geometric rotation to be applied, interpolation limits, etc…
- Once the value above is computed, it must be compared against the configurable threshold, and if the limit is exceeded the L1c Sunglint area flag is set for that particular measurement and pixel.

The flagging of the RFIs has two components – the one inherited from L1b/L1a and the one computed dynamically in L1c. For that, the RFIs seen in the current snapshot are retrieved from the AUX_RFI ADF and stored in a smaller variable, the rfi_pixel_list. Then, for each RFI in that list:

- The RFI position in (ξ, η) coordinates are computed as (ξ_{RFI}, η_{RFI}). If the source appears in FOV, a search is done in the 9 neighbours of the pixel corresponding to the source and the maximum value is then used as $T^p_{B,RFI}$
- If $T^p_{B,RFI}$ is greater than the configured minimum RFI BT threshold, all pixels contained in a circle of radius given by Eq. 57 and centred on (ξ_{RFI}, η_{RFI}) are marked as affected by RFI.

$$r = a\log\left(T^p_{B,RFI}\right)+b \qquad \textit{Eq. 57}$$

Note that this circle is given in pixel units, which corresponds to 2/128 in (xi, eta) coordinates. The values a and b are configurable (in the L1OP CNF file).

- In addition, if the $T^p_{B,RFI}$ is also greater than the threshold for tail flagging in the L1OP configuration file, the tails of that source are to be flagged as well. Eq.55 is used to compute the distance between the pixel (ξ, η) and the RFI (ξ_{RFI}, η_{RFI}). If the distance is smaller than the diameter of the circle previously computed, then that pixel is marked with the RFI tails flag. The tails have a width of $0.4r$ (r is computed in the Eq. above)

- Note: the RFI tails flag is the reuse of the EAFFOV flag (since all pixels were marked with it).

- Using the RFI System response computed in L1b, each pixel is flagged according to the estimated level of contamination at its position.
  - This mechanism uses the RFI BT estimation described above, $T^{p}_{B,RFI}$.
  - It also uses the boresight RFI system response computed in L1b processing, $\widehat{T_{RFI}}$
  - It translates the boresight RFI response to the position of each of the sources in the snapshot by using the Fourier transform properties:

$$\widehat{T_{RFI_n}} = T^{p}_{B,RFI} * \widehat{T_{RFI_0}} * e^{-i2\pi(u*\Delta\xi_n + v\Delta\eta_n)} \qquad \textbf{Eq. 58}$$

  - And the final image of the RFI disturbances is the sum of all individually translated and scaled RFI responses:

$$\widehat{T_{RFI_{Final}}} = \widehat{T_{RFI_1}} + \widehat{T_{RFI_2}} + \cdots + \widehat{T_{RFI_n}} \qquad \textbf{Eq. 59}$$

  - The expected contamination is computed by binning the real component of the inverse Fourier transform of the RFI impulse response frequencies, as shown in Figure 14[1]:

$$B_{RFI_{contamination}} = real(IFFT\{\widehat{BT_{RFI_{Final}}}\} \qquad \textbf{Eq. 60}$$

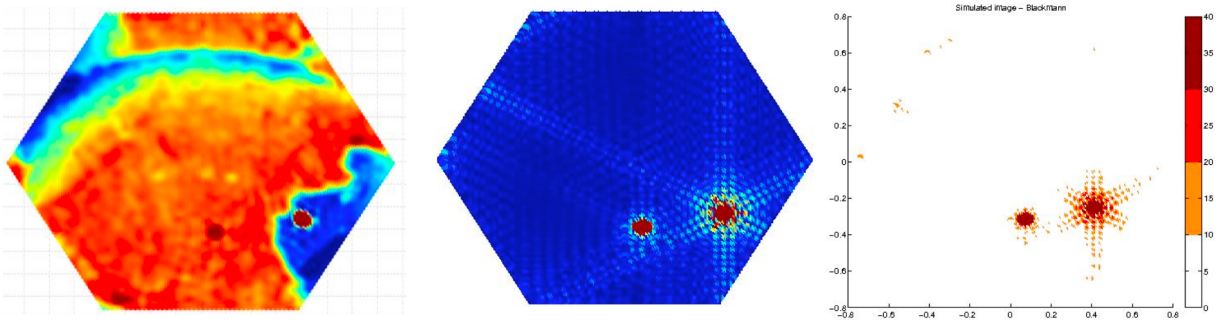

Figure 14: RFI Contamination flag estimation. After isolating the RFIs in the original image (left) and computing their intensity, a theoretical response is computed in L1b (middle). In L1c, each pixel position is compared to the temperature in the estimated RFI response and, depending on the level of contamination, flagged in 1 of 4 levels.

[1] In the case of HV polarisation, both T3 and T4 will have the same RFI flags. The contribution is computed using only the impulse response of a source with T3=1K and T4=0. Then, for each snapshot, the list of RFI present at the RFILST will be scanned for both T3 and T4 images to check if they are active. If they are active in at least one of T3 or T4, the $T^{p}_{B,RFI}$ for that particular source will be as the max(abs([SFT3, SFT4])). Note that the $T^{p}_{B,RFI}$ can be positive or negative, as derived from the T3 images or the T4 images, that is why we use Absolute first. The final impulse response accumulation is then used for both T3 and T4. This is done this way for simplicity, since the impulse response of Real of T3 is almost the same as Imaginary of T4

5.12.5. Error Handling

No errors should be returned on this function, unless for some strange reason two consecutive points in the projected EAF-FOV are the same, which would cause a division by zero in the border flag computation. However, this would indicate a previous error, not an error in the current processing.

If this situation happens, the pixel processing should be aborted, although the remaining pixels in the scene can still be processed.

5.13. Product Generation

L1c products shall be generated in accordance to the format described in [RD.5]. In this section it is described how to fill each substructure within each product type.

5.13.1. MPH

See section 4.1.1 of [RD.5] for the generic description. All fields in the MPH are applicable to all L1c products and are filled according to the following rules:

- Product: Filename of the physical file generated according to the rules in section 3.1
- Proc_Stage_Code: Taken from the configuration mode in which the L1OP is operating
- Ref_Doc: Hardcoded in the L1OP software to reflect which reference document is applicable
- Header_Size: Set to 9397
- Validation_Schema: Set by the Header template function of the BinXML function
- Acquisition_Station: Taken from the configuration file of L1OP
- Proc_Centre: Hardcoded to “L1OP”
- Proc_Time: Set to the UTC time of product generation
- Proc_Version: Harcoded in the L1OP software to reflect processor version
- Sensing_Start: UTC time of the first snapshot inside the Measurement Data Set
- Sensing_Stop: UTC time of the last snaphsot inside the Measurement Data Set
- Abs_Orbit: Absolute orbit at Sensing_Start, extracted from EE CFI function xo_time_to_orbit
- Phase, Cycle, Rel_Orbit: Values taken from EE CFI function xo_orbit_rel_from_abs
- X_Position, Y_Position, Z_Position, X_Velocity, Y_Velocity, Z_Velocity: Values taken from EE CFI function xo_propag
- Vector_Source: Depending on the data used for initialising the propagators it may be IG (propagation initialised with GPS data in ancillary packet), SP (propagation initialised with SOGS Predicted data) or SR (propagation initialised with SOGS Restituted data)
- Leap_UTC: Value retrieved from the EE CFI function xl_time_get_leap_second_info
- Product_Confidence: Set to NOMINAL if no errors occur during processing, or ERROR if errors 2-5 occur.
- Total_Size: Set to the binary product size plus the Header size

5.13.2. SPH

The generic format of L1c products SPH is defined in section 4.12, 4.5.1.2 and 4.5.2.2 of [RD.5].

The SPH is different between nominal L1c products and Browse L1c products. The only difference, however, lies in the field Incidence_Angle that is set for Browse products to indicate which incidence angle was taken to extract the swath.

All fields in the SPH are applicable to all L1c products and are filled according to the following rules:

- Abs_Orbit_Start, Start_Time_ANX_T: Values extracted from EE CFI function xo_time_to_orbit using MPH Sensing_Start as input
- Abs_Orbit_Stop, Stop_Time_ANX_T: Values extracted from EE CFI function xo_time_to_orbit using MPH Sensing_Stop as input
- UTC_at_ANX: Value extracted from EE CFI function xo_orbit_to_time using Abs_Orbit_Start and zero seconds as input
- Long_at_ANX: Value extracted from EE CFI function xo_propag_extra using Abs_Orbit_Start and zero seconds as input
- Ascending_Flag: Set to A (Ascending) if Abs_Orbit_Stop > Abs_Orbit_Start, or D (Descending) if Abs_Orbit_Stop = Abs_Orbit_Start
- Mode: Adopts different values depending on the file type.
  - (APID_DUAL) MIR_SCLD1C, MIR_SCSD1C, MIR_BWLD1C, MIR_BWSD1C
  - (APID_FULL) MIR_SCLF1C, MIR_SCSF1C, MIR_BWLF1C, MIR_BWSF1C
- Start_Lat, Start_Long: Set to boresight coordinates of first snapshot within product
- Stop_Lat, Stop_Long: Set to boresight coordinates of last snapshot within product
- Incidence_Angle: Set only for Browse products. Equal to 42.5º
- L1C_Err_Flag: Value set to 1 if L1C_Quality > L1C_Err_Threshold. Set to 0 otherwise
- L1C_Quality: Cumulative quality check set through the processing of the product.
- L1C_Err_Threshold: Value taken from L1c configuration file
- Apodisation_Window: Set to value according to apodisation ADF used
- RSC_Flag: Flag taken from original L1B products

5.13.3. Data Set

The product format variables are filled according to the list of variables presented in the previous chapters. The following sections detail which are the elements that shall be set as output.

5.13.3.1. Swath_Snapshot_list

- Snapshot_Time, Snapshot_ID, Snapshot_OBET: Values taken from l1b_data structure

- X_Position, Y_Position, Z_Position, X_Velocity, Y_Velocity, Z_Velocity, Vector_Source: Values taken from pvt_data structure.
- Q0, Q1, Q2, Q3: Values taken from aocs_data structure
- TEC: Value taken from tec variable
- Geomag_F, Geomag_D, Geomag_I: Values taken from geomag_f, geomag_d and geomag_I variables
- Sun_RA, Sun_DEC: Values taken from EE CFI function xl_sun and
- Sun_BT: Taken from l1b_data structure
- Accuracy: Taken from l1b_data structure
- Radiometric_Accuracy: Taken from bt_accuracy structure computed over the boresight direction

5.13.3.2. Temp_Swath_Dual

- Grid_Point_ID: Taken from earth_pixel_list structure
- Counter: Number of measurements existing for the previous particular Grid Point
- Flags: Taken from l1c_flags structure
- BT_Value: Taken from l1c_data structure
- Radiometric_Accuracy: Taken from bt_accuracy structure
- Incidence_Angle, Azimuth_Angle: Taken from pixel_observation_angles structure
- Faraday_Rotation_Angle, Geometric_Rotation_Angle: Taken from ionospheric_rotation_angles structure
- Snapshot_ID: Taken from l1b_data structure
- Footprint_Axis1, Footprint_Axis2: Taken from footprint_size structure

5.13.3.3. Temp_Swath_Full

Same as the previous case, except that the BT_Value is replaced by two different values:

- BT_Value_Real: Taken from l1c_data structure
- BT_Value_Imag: Taken from l1c_data structure

5.13.3.4. Temp_Browse

Same as Temp_Swath_Dual, except that in this case Incidence_Angle does not need to be defined

The computational strategy to generate the browse data is the following:

- Browse data are generated for an incidence angle of 42.5º
- For any given pixel, all measurements within ±5º are used to generate the interpolated value
- Only pixels which have measurements both above and below 42.5º are considered (in order to prevent extrapolation)

- Interpolation is done through a Levemberg-Marquardt LSQ fitting of all measurements inside the ±5º range
- LSQ interpolation is performed for the following values:
  - BT_Value_Real
  - BT_Value_Imag
  - Radiometric_Accuract
  - Azimuth_Angle (taking care to unwrap the phase to avoid interpolation jumps between 0 and 360º)
  - Footprint_Axis1 and Footprint_Axis2
- The information for the measurement flag is obtained from a bitwise OR operation on all the flags from the measurements within the ±5º range

6. OPEN ISSUES

- The computation of the pixel footprint through the analytical method described in Sections 5.10.1.2 and 5.10.4.2 can only be performed accurately for the case of the standard Blackman apodisation window. If the user chooses to use any other apodisation window, then the footprint value must be computed precisely, as described in Sections 5.10.1.1 and 5.10.4.1, by setting to false the Analytical Footprint flag in the L1-configuration file (CNF file) and activate the Compute footprint size precisely in the L1OP configuration file.