DESIGN DEFINITION FILE

BIBLOS

Prepared by: GMV

Approved by: Cristina de Negueruela

Authorized by: Lucia Soto

Code: GMV-BIBLOS-DD

Internal Code: GMV 21607/15 V1/15

Version: 1.0

Date: 15/04/2016
1. CHANGE RECORD

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Section</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>15/04/2016</td>
<td></td>
<td>Initial version for the BIBLOS web</td>
</tr>
</tbody>
</table>
# 2. TABLE OF CONTENTS

1. CHANGE RECORD .................................................................................................................. 2
2. TABLE OF CONTENTS ............................................................................................................. 3
3. LIST OF TABLES AND FIGURES .......................................................................................... 6
4. INTRODUCTION ..................................................................................................................... 9
   4.1. PURPOSE ............................................................................................................................ 9
   4.2. SCOPE ............................................................................................................................... 9
5. REFERENCES .......................................................................................................................... 10
   5.1. APPLICABLE DOCUMENTS ............................................................................................. 10
   5.2. REFERENCE DOCUMENTS ............................................................................................... 10
   5.3. PROJECT DOCUMENTS ................................................................................................... 11
   5.4. DEFINITIONS ................................................................................................................ 12
       5.4.1. PIXEL .......................................................................................................................... 12
       5.4.2. SUBPIXEL ................................................................................................................... 12
       5.4.3. SCALE FACTOR .......................................................................................................... 12
       5.4.4. MEMORY MANAGEMENT - SQUARES ...................................................................... 12
       5.4.5. SQUARE SIZE BEFORE AND AFTER THE SPATIAL RESAMPLING (INSTRUMENT MODULE) ................................................................................................................... 13
       5.4.6. SQUARE MARGIN ....................................................................................................... 15
       5.4.7. RELATION BETWEEN SQUARE INPUT/OUTPUT SQUARE SIZE, SQUARE MARGIN, SCALE FACTOR AND NUMBER OF DETECTOR ELEMENTS ......................................................... 16
       5.4.8. NUMBER OF SQUARES ACT AND ALT ..................................................................... 17
       5.4.9. NOMINAL, SATELLITE AND RESTITUTED ORBIT AND ATTITUDE .......................... 17
6. GEOMETRY MODULE.............................................................................................................. 19
   6.1. ORBIT BLOCK .................................................................................................................. 19
       6.1.1. DETAILED DESCRIPTION ....................................................................................... 21
       6.1.2. FLOW DIAGRAM ...................................................................................................... 22
       6.1.3. INTERFACES ............................................................................................................ 23
       6.1.4. SCOPE AND LIMITATIONS ...................................................................................... 24
       6.1.5. EXTERNAL LIBRARIES ............................................................................................ 24
       6.1.6. ERRORS AND WARNINGS ...................................................................................... 25
   6.2. ATTITUDE BLOCK .......................................................................................................... 25
       6.2.1. DETAILED DESCRIPTION ....................................................................................... 27
       6.2.2. FLOW DIAGRAM ...................................................................................................... 31
       6.2.3. INTERFACES ............................................................................................................ 33
       6.2.4. SCOPE AND LIMITATIONS ...................................................................................... 34
       6.2.5. EXTERNAL LIBRARIES ............................................................................................ 34
       6.2.6. ERRORS AND WARNINGS ...................................................................................... 34
   6.3. AOCS/INSTRUMENT COUPLING BLOCK ......................................................................... 34
       6.3.1. DETAILED DEFINITION ............................................................................................ 37
       6.3.2. FLOW DIAGRAM ...................................................................................................... 42
       6.3.3. INTERFACES DEFINITION ....................................................................................... 44
       6.3.4. SCOPE AND LIMITATIONS ...................................................................................... 47
       6.3.5. EXTERNAL LIBRARIES ............................................................................................ 47
       6.3.6. ERRORS AND WARNINGS ...................................................................................... 47
   6.4. SCENE INTERACTION BLOCK .......................................................................................... 47
       6.4.1. DETAILED DEFINITION ............................................................................................ 54
       6.4.2. FLOW DIAGRAM ...................................................................................................... 59
13. ANNEX. REFERENCE FRAMES AND TRANSFORMATIONS................................................................. 107
  13.1. ECEF ......................................................................................................................................... 107
  13.2. ECI ............................................................................................................................................. 107
  13.3. ECEF <-> ECI TRANSFORMATION ......................................................................................... 107
  13.4. GEODETIC REFERENCE FRAME ............................................................................................... 107
  13.5. ECEF TO GEODETIC TRANSFORMATION .............................................................................. 108
  13.6. GEODETIC TO ECEF TRANSFORMATION .............................................................................. 108
  13.7. RESTITUTED ORBITAL REFERENCE FRAME ............................................................................ 108
  13.8. SATELLITE REFERENCE FRAME .............................................................................................. 108
  13.9. INSTRUMENT REFERENCE FRAME ......................................................................................... 109
  13.10. ORBITAL TO ECEF AND ECI TRANSFORMATIONS ............................................................. 109
  13.11. EULER ANGLES, ROTATION MATRIX AND QUATERNIONS CONVERSIONS.................... 110
  13.12. UNIVERSAL TRANSVERSE MERCATOR COORDINATE SYSTEM (UTM) ......................... 110
14. ANNEX. TIME REFERENCES .......................................................................................................... 113
  14.1. MODIFIED JULIAN DAY (MJD2000), EPOCH .......................................................................... 113
  14.2. JULIAN DAY (JD) ..................................................................................................................... 113
  14.3. UNIVERSAL TIME (UT) ........................................................................................................... 113
  14.4. GREENWICH (GMT) ................................................................................................................ 113
  14.5. COORDINATED UNIVERSAL TIME (UTC) ............................................................................. 113
  14.6. INTERNATIONAL ATOMIC TIME (TAI) ................................................................................... 113
  14.7. LEAP SECONDS ........................................................................................................................ 114
  14.8. TIME CONVERSIONS .............................................................................................................. 114
15. ANNEX. INTERPOLATION METHOD ............................................................................................... 115
3. LIST OF TABLES AND FIGURES

Table 5-1 Applicable Documents .................................................................10
Table 5-2 Project documents ......................................................................12
Table 5-3 Orbit File Formats. Source: ESA ...............................................17
Table 5-4 Attitude File Formats. Source: ESA ...........................................17
Table 5-5 Usage of Orbit and Attitude Files in E2ES. Source: ESA ..........18
Table 6-1 Configuration Parameters definition for orbit block .................24
Table 6-2 Attitude Block Input Parameters ..................................................24
Table 6-3 Attitude Block Output Parameters .............................................24
Table 6-4 Configuration Parameters definition for attitude block ..........33
Table 6-5 Attitude Block Input Parameters ..................................................33
Table 6-6 Attitude Block Output Parameters ..........................................33
Table 6-7 Configuration Parameters definition template example ..........45
Table 6-8 Inputs definition ......................................................................46
Table 6-9 Outputs definition ......................................................................46
Table 6-10 Configuration Parameters definition template example .......61
Table 6-11 Inputs definition ......................................................................62
Table 6-12 Outputs definition ...................................................................62
Table 7-1 Configuration Parameters definition for attitude block ..........69
Table 7-2 Attitude Block Input Parameters ................................................69
Table 7-3 Attitude Block Output Parameters ............................................69
Table 7-4 Configuration Parameters definition for attitude block ..........75
Table 7-5 Attitude Block Input Parameters ................................................75
Table 7-6 Attitude Block Output Parameters ............................................75
Table 8-1 Configuration Parameters definition template example ..........89
Table 8-2 Inputs definition ......................................................................90
Table 8-3 Outputs definition ...................................................................90
Table 8-4 Configuration Parameters definition template example ..........98
Table 8-5 Inputs definition ......................................................................99
Table 8-6 Outputs definition ...................................................................99
Table 10-1 List of EOCFI Classes used and use in BIBLOS. .................102

Figure 5-1: Memory Management - Division of the Scene in Squares ..........13
Figure 5-2: Spatial resampling from Subpixel to Pixel ..............................14
Figure 5-3: Size of the Square before and after the spatial resampling ....15
Figure 5-4: Square Margin .....................................................................16
Figure 6-1: Geometry Module ................................................................19
Figure 6-2 Orbit Block, Satellite & Restituted Orbit Simulation ............23
Figure 6-3: Geodetic and geocentric pointing attitude .............................25
Figure 6-4: Diagram representing the effect of Yaw Steering correction on the acquisition image. Left – no yaw steering. Right – with yaw steering ........................................26
Figure 6-5 Attitude Block, Nominal, Satellite & Restituted Attitude generation ...................................................32
Figure 6-6: Generation of LOS direction (in camera frame), knowing location at the focal plane and focal length .................................................................35
Figure 6-7: Pushbroom. Source http://wlab.iis.u-tokyo.ac.jp/ .................36
Figure 6-8: Detector line. Every instrument acquisition time, each detector element images one pixel.
.................................................................................................................................................. 36
Figure 6-9: Detector line simulation at subpixel level. The detector line is artificially widened a Square
Margin number of subpixels ............................................................................................................. 37
Figure 6-10: Location of the first subpixel. The first subpixel is in the Square Margin, and has X<0 and
Y>0 coordinates. ............................................................................................................................ 38
Figure 6-11: Location of the centre position of the detector line when the number of subpixels across
track is ODD, is in the centre of a subpixel.................................................................................... 39
Figure 6-12: Location of the centre position of the detector line when the number of subpixels across
track is EVEN, is between two subpixels. ....................................................................................... 39
Figure 6-13: Position of a subpixel (i,j) w.r.t. the position of the first subpixel. ............................ 40
Figure 6-14: Generation of LOS direction (in camera frame), knowing location at the focal plane and
focal length .......................................................................................................................................... 40
Figure 6-15: Projection of the LOS to ECEF and Acquisition Area ................................................. 41
Figure 6-16 Calculation of the LOS in instrument frame ................................................................. 43
Figure 6-17 Projection of the LOS in ECEF ..................................................................................... 44
Figure 6-18: Acquisition with DEM ............................................................................................... 48
Figure 6-19: Acquisition with DEM ............................................................................................... 49
Figure 6-20: Iteration to find the intersection of the LOS with the DEM. .................................... 49
Figure 6-21: Case 1: Acquisition in a flat terrain profile ................................................................. 50
Figure 6-22: Case 2: Acquisition in an ascending terrain profile .................................................... 50
Figure 6-23: Case 2: Acquisition in a descending terrain profile .................................................... 51
Figure 6-24: Case 2: Acquisition in a descending terrain profile with a steep slope .................... 51
Figure 6-25: Observation zenith angle, Sun zenith angle ............................................................... 52
Figure 6-26: Relative Azimuth Angle ............................................................................................. 53
Figure 6-27: Typical pointing for a limb-sounding mission. Source: EOCFI Pointing Library
documentation, [RD.26] .................................................................................................................. 54
Figure 6-28: Scene Interaction Block memory management .......................................................... 54
Figure 6-29: Scene Interaction Block architecture and memory management ............................... 59
Figure 6-30: Scene Interaction Block flow diagram ........................................................................ 61
Figure 7-1: Scene Generation Module ........................................................................................... 64
Figure 7-2: Resampling Block ......................................................................................................... 65
Figure 7-3: Bilinear interpolation. Source RD.14] ....................................................................... 66
Figure 7-4 Resampling block flow diagram .................................................................................... 68
Figure 7-5 Atmosphere simulator block flow diagram ................................................................. 74
Figure 8-1: Instrument Module ....................................................................................................... 77
Figure 8-2: Memory Management - Division of the Scene in Squares .......................................... 79
Figure 8-3: Double loop in the ACT and ALT directions to process each square ....................... 79
Figure 8-4: Reading the input Scene and assigning the information to each square. Example for the
first square .......................................................................................................................................... 80
Figure 8-5: Characterization of spectral bands of a reference image (dotted black lines) with respect to
some of the typical imager bands (multispectral blue, green, red, panchromatic)......................... 81
Figure 8-6: Example MTF 1D defined in the inputs, with the value at the Nyquist frequency for the
detector (detector pitch (pixel size) = 52e-6m), and at subpixel level (subpixel size = pixel size/3,
as the scale factor is 3). ..................................................................................................................... 83
Figure 8-7: MTF in 2D calculated from the MTFs in 1D for the across and along track dimensions. The
MTF in 2D is only calculated for 1024x1024 samples of subpixels because that is the size of image
processed in each step. The MTF in 1D for the ACT and ALT directions in this example are the
same, so in this case the MTF2D is symmetrical in the X and Y axis. ............................................ 84
Figure 8-8: Size of the Square before and after the spatial resampling .......................................... 86
Figure 8-9: Writing image to file. After a row of Squares is processed it is written to file ............. 89
Figure 8-10 Spatial block flow diagram .......................................................................................... 89
Figure 8-11 Radiometric block flow diagram ................................................................. 97
Figure 10-1: Orbital Frame in EOCFI. Source: EOCFI Conventions Document, [RD.28] .......... 103
Figure 10-2: Orbital Frame in EOCFI. Source: EOCFI Pointing Library, [RD.26] ...................... 103
Figure 13-1: Orbital Frame ......................................................................................... 108
Figure 13-2: Instrument Frame, X-Y plane (=focal plane) .................................................... 109
Figure 13-3: Instrument Frame ....................................................................................... 109
Figure 13-4: UTM projection .......................................................................................... 110
Figure 13-5: UTM Grid ................................................................................................. 111
Figure 13-6: UTM Zone 30 (containing Spanish territory) .................................................. 111
4. INTRODUCTION

4.1. PURPOSE

This document describes the list of blocks implemented in this contract for the BIBLOS project. The blocks constitute a basic instrument data generator for a passive optical instrument. These blocks are part of the Geometry, Scene Generator and Instrument modules. The blocks are:

- Geometry Module:
  - Orbit block
  - Attitude block
  - AOCS/Instrument Coupling block
  - Scene Interaction block

- Scene Generation Module
  - Resampling block
  - Atmosphere block

- Instrument Module
  - Spatial block
  - Radiometric block.

Each of the following sections describes in detail the architecture of each block. Each section contains: the detailed description, flow diagram, interfaces definition, scope and limitations, Use of External Libraries if applicable, revision history and the requirements it meets.

4.2. SCOPE

This document is performed under ESA contract 4000112456/14/NL/FF/gp. This project is carried out by a consortium led by GMV and including the different members:

- GMV, responsible for management, system-level activities and active optical instruments.
- Warsaw Military University of Technology, consultant for passive optical instruments.
5. REFERENCES

5.1. APPLICABLE DOCUMENTS

The following documents, of the exact issue shown, form part of this document to the extent specified herein. Applicable documents are those referenced in the Contract or approved by the Approval Authority. They are referenced in this document in the form [AD.X]:

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
<th>Code</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[AD.3]</td>
<td>Minutes of the Meeting of the PCR</td>
<td>GMV-BIBLOS-MOM-005-PCR_2015_03_19_v1.0</td>
<td>1.0</td>
<td>19/03/2015</td>
</tr>
</tbody>
</table>

Table 5-1 Applicable Documents

5.2. REFERENCE DOCUMENTS

The following documents, although not part of this document, amplify or clarify its contents. Reference documents are those not applicable and referenced within this document. They are referenced in this document in the form [RD.X]:

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
<th>Code</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RD.1]</td>
<td>EO Missions and Elements Categorization</td>
<td>ARCHEO-E2E-TN-001</td>
<td>2.0</td>
<td>19/09/2012</td>
</tr>
<tr>
<td>[RD.9]</td>
<td>Spatial and Spectral Effects of BIBLOS</td>
<td>Memo</td>
<td></td>
<td>16/04/2015</td>
</tr>
<tr>
<td></td>
<td><a href="http://www.cplusplus.com/doc/tutorial/variables/">http://www.cplusplus.com/doc/tutorial/variables/</a>. 18/05/2015</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://modtran5.com/">http://modtran5.com/</a>. 12 March 2015</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 5-1 Reference Documents

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
<th>Code</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RD.23]</td>
<td>IAU Standards of Fundamental Astronomy Board, The SOFA software libraries</td>
<td></td>
<td>11</td>
<td>2015/02/09</td>
</tr>
</tbody>
</table>

### 5.3. PROJECT DOCUMENTS

The following documents are produced in the frame of this activity. They are referenced in this document in the form [PD.X]:

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Title</th>
<th>Code</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[PD.1]</td>
<td>EO models library roadmap</td>
<td>GMV-EOM-RM</td>
<td>3.0</td>
<td>25/02/2016</td>
</tr>
<tr>
<td>[PD.3]</td>
<td>Phase 2 technical proposal</td>
<td>GMV-EOM-PR</td>
<td>1.3</td>
<td>26/06/2016</td>
</tr>
</tbody>
</table>
5.4. DEFINITIONS

For additional definitions please refer to [PD.2]. In this section definitions that are related to this document are included.

5.4.1. PIXEL

A pixel is the area imaged by one detector element for one instrument acquisition time. Each pixel is stored as one value in the image captured by the satellite.

5.4.2. SUBPIXEL

For the instrument modelisation, to apply the spatial aberrations, for the forward model and for adjacency purposes, the scene is simulated at subpixel level. This means that each pixel is artificially divided into a number of subpixels determined by the scale factor.

5.4.3. SCALE FACTOR

The pixel shall be divided into a number of pixels equal to the square of the scale factor. If the resampling factor is 3 for example, there are 9 subpixels per pixel imaged.

5.4.4. MEMORY MANAGEMENT - SQUARES

The scene to be acquired for the Instrument Data Simulator, and the Processing, is typically in the tens of kilometres range. For average detector pitch (pixel size), and resolution, this is a huge file size. For example, if we are processing a square image of 60km, and the ground sampling distance (distance each pixel is imaging on Earth) is 10m (spatial resolution). For example, if each pixel is saved in 1 bytes (that is equivalent to 8 bits, or 256 radiometric levels of resolution). The size of the scene for each band is \((60e3/10\text{pixels})^2\times1\text{byte}=34\text{Mb.}\) Additionally you need to keep in memory during the processing the input image to the instrument at subpixel level (which is usually in double precision and to the square of the scale factor). Without counting the Square Margin, for a scale factor of 3, the size of the Scene at the entrance of the Instrument Model with the previous example is \((60e3*5f(3)/10)^2*8\text{bytes}=2.4\text{Gb.}\) This is not feasible for usual working stations.

To overcome this issue the Scene is divided and processed in units that shall be named in the BIBLOS project as **Squares**. A Square is a portion of the area to be imaged, the Scene. Each Square is processed independently and then saved to file to form the full Scene.
5.4.5. SQUARE SIZE BEFORE AND AFTER THE SPATIAL RESAMPLING (INSTRUMENT MODULE)

One of the complex issues of the instrument modelling, memory management and configuration is that in the spatial resampling the subpixels are averaged into pixels. The spatial resampling takes place in the Spatial block in the Instrument Module after the spatial aberrations are applied. In the following figure it is shown for each Square how the Spatial resampling changes the Square from subpixels to pixels. That is, the Line-of-sight is generated for each subpixel, the Scene Generator works at subpixel level and calculates the TOA for each subpixel, the TOA is then processed in the Spatial Block where it is spatially resampled into pixels. After that the image is processed at pixel level (Radiometric effects, Image in Digital numbers). The following figure shows the spatial resampling:

<table>
<thead>
<tr>
<th>Square 1,1</th>
<th>Square 1,2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square 2,1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Square i,j</td>
</tr>
</tbody>
</table>

**Figure 5-1: Memory Management - Division of the Scene in Squares**
To handle in memory each square the size of the Square before and after spatial resampling is calculated. There are several configuration parameters that are related:

- **Input Square Size** – Size of the Square before spatial resampling
- **Output Square Size** – Size of the Square after spatial resampling
- **Scale Factor** – factor by which the pixel is divided into subpixels, see detailed description above.
- **Square Margin** – Extra number of subpixels simulated around the Square that are cut during the spatial resampling, see detailed description above.

These 4 configuration parameters need to be such that is possible during the spatial resampling the Square Margin is cut off, the subpixels are averaged into pixels. The exact relation is:

\[
\text{Input square size} - 2 \times \text{square margin} \div \text{scale factor} = \text{output square size}
\]

Additionally a second necessary condition is that the output square size must be a multiple of the number of detector elements.

\[
\text{remainder of (number of detector elements / output square size)} = 0
\]
5.4.6. SQUARE MARGIN

Each Square of image processed needs an excess margin of image that will be cut in the spatial block. This area is needed for adjacency calculations and for border effects. This extra area is called the Square Margin.

Figure 5-3: Size of the Square before and after the spatial resampling
5.4.7. RELATION BETWEEN SQUARE INPUT/OUTPUT SQUARE SIZE, SQUARE MARGIN, SCALE FACTOR AND NUMBER OF DETECTOR ELEMENTS.

The scale factor, square input and output size and the scale factor are configurable parameters. These parameters are dependent and must verify the following equation. If they do not comply the simulation will stop, it is a necessary condition.

\[(\text{Input square size} - 2 \times \text{square margin}) / \text{(scale factor)} = \text{output square size}\]

Additionally a second necessary condition is that the output square size must be a multiple of the number of detector elements.

\[\text{remainder of (number of detector elements / output square size)} == 0\]

For example, if scale factor = 3; square margin = 62; input square size = 1024, input square size = 300 and 1500 detector elements in the detector line.

1. The following relationship is met: \((1024 - 2 \times 62) / 3 = 300\)
2. \(1500 / 300 = 5\). The remainder of 1500/300 is 0.

An example that does not work: if the detector line has 2000 elements, the output block size cannot be 300 (remainder of \((2000 / 300 = 200)\)). The output block size could be 500 elements.

An example that does not work: In the first example the input block size could not be 1000 because \((1000 - 2 \times 62) / 3 \neq 300\)

There are limited combinations of the input and output Square size, Square Margin, Scale Factor and number of detector elements.
5.4.8. NUMBER OF SQUARES ACT AND ALT.

The number of squares across track is the number of detector elements divided by the output square size (output is after spatial resampling).

\[ n\_sq\_ACT = \frac{n\_pix}{output\_square\_size} \]

For example, if there are 1500 detector elements, and the square size is 300 pix x 300 pix, there are 5 blocks across track.

Along track this depends on the scene size, so on the orbit. The number of squares along track is the floor of the number of Satellite orbit positions (each orbit position represents an acquisition time), divided by the output block size.

\[ n\_sq\_ALT = \text{floor}(\frac{\text{size(Satellite\_orbit)}}{output\_square\_size}). \]

5.4.9. NOMINAL, SATELLITE AND RESTITUTED ORBIT AND ATTITUDE.

The Orbit and Attitude files can take several formats, that correspond to different analyses and processes. For example a predicted orbit file, and propagated orbit file, the result of the navigation, or the guidance. BIBLOS follows the EOCFI naming convention, [RD.26]. In sections §5.4.9.1 and §5.4.9.2 a list of file formats is shown, and the usage in the domain of the end-to-end simulators.

BIBLOS uses the following Orbit and Attitude files:

- **Nominal Attitude**: This is the ideal orientation of the platform. The law of attitude is calculated (geodetic pointing, yaw steering, etc).
- **Satellite Orbit & Attitude**: This is the real trajectory of the satellite and the real orientation.
- **Restituted Orbit & Attitude**: This is the knowledge of trajectory and orientation. This comes from the GNSS sensors, the ranging measurements, and the sensors on board the spacecraft.

5.4.9.1. Orbit and Attitude Files Formats used in Earth Observation

A description of the common Orbit and Attitude files is presented in the two tables below (source ESA):

<table>
<thead>
<tr>
<th>Orbit Files</th>
<th>Type</th>
<th>Format Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orbit Scenario File (OSF)</td>
<td>Theoretical: Reference Orbit definition (repeat cycle, cycle length MLST)</td>
<td>EO-MA-DMS-GS-0007 v4.10, [RD.26]</td>
</tr>
<tr>
<td>TLE</td>
<td>Theoretical: Orbital elements</td>
<td><a href="http://celestrak.com/">http://celestrak.com/</a></td>
</tr>
<tr>
<td>Restituted Orbit File (ROF)</td>
<td>Discrete List of State Vectors (OSV)</td>
<td>EO-MA-DMS-GS-0007 v4.10, [RD.26]</td>
</tr>
<tr>
<td>Predicted Orbit File (POF)</td>
<td>Discrete List of State Vectors (OSV)</td>
<td><a href="https://igscb.jpl.nasa.gov/igscb/data/format/sp3_docu.txt">https://igscb.jpl.nasa.gov/igscb/data/format/sp3_docu.txt</a></td>
</tr>
<tr>
<td>SP3</td>
<td>Discrete List of State Vectors (OSV)</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5-3 Orbit File Formats. Source: ESA**

<table>
<thead>
<tr>
<th>Attitude Files</th>
<th>Type</th>
<th>Format Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attitude Definition File (ADF)</td>
<td>Theoretical: Attitude Law definition (Yaw Steering, Local Normal, Zero Doppler, etc) and optionally other initialization methods.</td>
<td>EO-MA-DMS-GS-0007 v4.10, [RD.26]</td>
</tr>
<tr>
<td>SP3</td>
<td>Discrete List of State Vectors (OSV)</td>
<td>EO-MA-DMS-GS-0007 v4.10, [RD.26]</td>
</tr>
</tbody>
</table>

**Table 5-4 Attitude File Formats. Source: ESA**
5.4.9.2. Usage of Orbit and Attitude Files within E2E Simulation Chains

A description of the use of the Orbit and Attitude files in the end-to-end simulators is presented in the table below (source ESA):

<table>
<thead>
<tr>
<th>Role Description</th>
<th>Orbit</th>
<th>Attitude</th>
<th>Earth Observation Cfi Name</th>
<th>Comments/examples</th>
<th>File Formats (Orbit)</th>
<th>File Formats (Attitude)</th>
</tr>
</thead>
</table>
| The reference orbit and attitude according to the mission definition             | X     | X        | NOMINAL                     | - Basis to compute position and attitude according to the ideal mission definition.  
- Used for Mission Analysis activity to check suitability of a specific reference orbit to satisfy mission needs, or coverage.  
- Used to define an orbit/attitude identical to the reference (e.g. when nothing else is available) | - Orbit Scenario File (OSF)  
- TLE  
- List of OSV: Long term Predicted File (special case from ESOC) | - Attitude Law:  
Attitude Definition File (ADF)                                                                 |
| The real orbit and attitude of the satellite. It represents the physical truth.  | X     | X        | SATELLITE                   | - As good as the ability of the satellite to actually achieve reference orbit and attitude.  
- Can be based on a reference orbit and attitude in ideal case.  
- Can be based on Orbit prediction | - Orbit Scenario File (OSF)  
- TLE  
- List of OSV: Predicted Orbit File (POF) | - Attitude Law:  
Attitude Definition File (ADF)  
- List of Quaternion: Attitude file (ATT) |
| The knowledge of the orbit and attitude of the satellite.                        | X     | X        |                             | Can be:  
- Based on measured position and attitude (on-board e.g. GNSS) or on-ground (tracking)  
- Based on an estimated position and attitude (e.g. via SW) or POD system (restituted).  
- Starting position for the instrument position and attitude  
- Used in the Data Processing of an E2E chain | - List of OSV:  
Restituted Orbit File (ROF)  
- List of Quaternion: Attitude file (ATT) |                                                                 |
| The real or known attitude of an instrument. It has a fixed or time dependent attitude expressed wrt the real or known satellite attitude. | X     |          | INSTRUMENT                   | - used for visibility/ pointing/ geolocation calculation.  
- can be computed respectively from real or from the known attitude of the satellite  
- Used in the Simulation or Data Processing of an E2E chain |                                                                 | - List of Quaternion: Attitude file (ATT) |

Table 5-5 Usage of Orbit and Attitude Files in E2ES. Source: ESA
6. GEOMETRY MODULE

The Geometry module architecture, and the relation between the blocks is shown in the figure below:

The geometry module is composed of 4 blocks: the orbit, attitude, AOCS/Instrument coupling and the scene interaction block. The description of each block is done in the following sections.

6.1. ORBIT BLOCK

The Orbit Block is in charge of the simulation of the satellite’s orbital motion. This block computes the Satellite orbit from the input orbit parameters. In addition, the Restituted orbit is calculated to
simulate the knowledge of the satellite of its position. The final outputs of the BB are Satellite Orbit and Restituted Orbit which will be used for the Attitude Simulator, the Scene Generator and the L1 processing.

**Satellite Orbit simulator**

The Satellite Orbit simulator is responsible for the detailed propagation of the orbit, considering all of the orbit perturbations selected by the user, in a selected timeframe from the initial orbit parameters provided by the input state vector (OSV or OSF file, see EOCFI documents for format). The simulation is conducted only over a short time period some time before and during the time when the scene is captured.

As an input Satellite Orbit Simulator takes:

- Initial Orbital State Vector (OSV file), or OSF file.
- Length along track of the scene
- Reference time (the initial state vector shall be propagated starting from this reference time).
- Perturbations flags selecting which perturbations will be considered (simulation parameter)
- Perturbations models Parameters – an configuration file listing all parameters describing perturbation models used
- Time step for propagation (instrument sampling time)

**Restituted Orbit simulator**

The Restituted Orbit Simulator simulates the knowledge of the satellite of its position and the process of orbit determination on board the satellite. First, this module adds an Error Function (specified by user, by an appropriate flag) to the Satellite Orbit Simulator data. The Error Function simulates the uncertainties of the SC positioning system. Then it interpolates the Satellite Orbit Simulator data to match the SC positioning system data acquisition rate.

**Orbit Error Function**

The resulting Restituted position and velocity is the vector sum of the state vector determined from the Satellite Orbit Simulator and the Error Function output. The Error Function (both for the position and for the velocity) is calculated in the three vector components: along-track, across-track and radial. This Error Function vector is then transformed to the coordinates of the State Vectors in ECEF frame.

There are a few Error Function forms that may be used:

- **Polynomial Orbit Error Function**
  Error Function for each vector component is described by the equation:
  
  \[ E_{rr} = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \]
  
  Where:
  - \( a \) are the polynomial coefficients, input by the user.
  - \( t \) is a time in seconds since the beginning of simulation
  - \( t_0 = (MJD2000_{ini} - MJD2000) \times 86400 \)
  - \( E_{rr} \) is module of the vector component of the Error Function

- **Sinusoidal Orbit Error Function**
  Error Function for each vector component is described by the equation:
\[ E_{rr} = a_0 \sin(2\pi a_1 t + \text{phase}) \]

Where:
- \( a_1 \) is the amplitude of the error, input by the user.
- \( a_2 \) is the frequency of the sinusoidal error, input by the user.
- \( t \) is a time in seconds since the beginning of simulation
- \( t_0 = (\text{MJD2000}_{\text{ini}} - \text{MJD2000}) \times 86400 \)
- Phase is the phase added to the sinusoidal model
- \( E_{rr} \) is module of the vector component of the Error Function

### 6.1.1. DETAILED DESCRIPTION

The following process is followed to generate the Satellite and Restituted Orbit.

#### Satellite Orbit simulator

1. “Initial State Vector” is taken as input from the OSF or OSF file (for the OSF file, the osvCompute function in the OrbitId class of the EOCFIs, internally and transparent to the user, calculated the initial state vector).

2. The step size is defined in the global configuration file, and it is the instrument sampling time. The number of steps is derived from the scene size in the along track direction (local geometry file configuration parameters). The number of steps is calculated with the following:

   \[
   \text{time}_{\text{acquisition\_s}} = \frac{\text{scene\_size\_alt}}{v} \times \left( \frac{r}{r_e} \right);
   \]

   \[
   \text{n\_sat\_orb\_pos} = \frac{\text{time}_{\text{acquisition\_s}}}{\text{instrument\_sampling\_time}};
   \]

   Where:
   - \( \text{time\_acquisition\_s} \) is the duration of the acquisition in seconds
   - \( \text{scene\_size\_alt} \) is the length of the scene in the along track direction in meters
   - \( v \) is the modulus of the satellite velocity in m/s
   - \( r \) the the satellite altitude in meters
   - \( r_e \) is the radius of the equator in meters

3. The resultant propagated orbit is then cut to the \(< t_{\text{start}}; t_{\text{end}} >\) range. The output is “Satellite Orbit”

#### Restituted Orbit simulator

4. Estimation Error Function is calculated in Orbital Reference Frame (radial, along- and across-track directions) based on the Error Function parameters provided by the user.

   1. In case of polynomial error function:

   \[
   E'_{rr} = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4
   \]

   Where:
- $a_i$ are polynomial coefficients, input by the user in the International System of units (metres, and metres divided by powers of seconds).
- $t$ is a time in seconds since the beginning of simulation
  $t_0 = (MJD_{2000_{ini}} - MJD_{2000}) \cdot 86400s$
- $E_{rr}^j$ is module of the $j$ vector component of the Error Function in metres and m/s.

$$E_{rr} = \begin{bmatrix} E_{rr}^{radial} \\ E_{rr}^{along} \\ E_{rr}^{across} \end{bmatrix}$$

2. In case of polynomial error function:

$$E_{rr}^j = a_0 \sin(2\pi a_1 t)$$

Where:
- $a_0$ is the amplitude of the Error Function, input by the user in the International System of units.
- $t$ is a time in seconds since the beginning of simulation
  $t_0 = (MJD_{2000_{ini}} - MJD_{2000}) \cdot 86400s$
- $E_{rr}^j$ is module of the $j$ vector component of the Error Function in metres and m/s.

$$E_{rr} = \begin{bmatrix} E_{rr}^{radial} \\ E_{rr}^{along} \\ E_{rr}^{across} \end{bmatrix}$$

3. In case of multiple Error Functions combined:

$$E_{rr} = (E_{rr})_{poly} + (E_{rr})_{sin} + \ldots$$

5. Error Function is projected onto the state vector frame (ECEF).

$$(E_{rr})_{ECEF} = R_{ECEF}^{Orb} \cdot (E_{rr})_{Orb}$$

6. Satellite orbit and Error Function projection are summed for the position:

$$r^{EST} = r^{Real} + (E_{rr})_{ECEF}$$

The same equation is applied to the velocity with the corresponding errors for the velocity.

The output is "Restituted Orbit"

### 6.1.2. FLOW DIAGRAM

The following diagram depicts the main steps of the orbital Building Block.
6.1.3. INTERFACES

6.1.3.1. Configuration Parameters

The following configuration parameters are defined.

- Global includes parameters that affect several modules
- Local refers to parameters that affect only this module
Variable Type is the C data type

Table 6-1 Configuration Parameters definition for orbit block

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene size along track</td>
<td>value describing the size of the scene in meters in the along track direction (the across track direction of the acquired image is determined by the detector line).</td>
<td>double</td>
<td>m</td>
<td>local</td>
</tr>
<tr>
<td>Instrument Sampling Time</td>
<td>Time rate at which instrument samples the image</td>
<td>Double</td>
<td>s</td>
<td>Global</td>
</tr>
</tbody>
</table>

6.1.3.2. Inputs

Table 6-2 Attitude Block Input Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial state vector</td>
<td>Initial position and velocity of the analysed SC in ECEF</td>
<td>Earth explorer file (OSV or OSF) with the state vector</td>
<td>Mjd2000, m, m/s</td>
</tr>
</tbody>
</table>

6.1.3.3. Outputs

Table 6-3 Attitude Block Output Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellite Orbit</td>
<td>Satellite orbit as calculated from the Orbit Block Size: Mx7, where M is the number of instants of time of the satellite ephemerides generation 7 elements of each row are: the epoch, 3 positions and 3 velocities in ECEF</td>
<td>Earth explorer file with list of ephemeris.</td>
<td>time, m, m/s</td>
</tr>
<tr>
<td>Restituted Orbit</td>
<td>Restituted orbit as calculated from the Orbit Block Size: Mx7, where M is the number of instants of time of the satellite ephemerides generation 7 elements of each row are: the epoch, 3 positions and 3 velocities in ECEF</td>
<td>Earth explorer file with list of ephemeris.</td>
<td>time, m, m/s</td>
</tr>
</tbody>
</table>

6.1.4. SCOPE AND LIMITATIONS

This block is applicable to all Earth Observation missions, including missions with the following instruments:

- Passive Opticals
- Active Microwaves
- Passive Microwaves
- Active Opticals

6.1.5. EXTERNAL LIBRARIES

The libraries used inside this block are:

- Eigen, §11.1
- NetCDF §11.5
- EOCFIs §11.6
6.1.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfiError Error class, [RD.26]. Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception in thrown in case of error and the execution stops. Warnings do not stop the execution.

For the Orbit block the following errors can occur:
- File access: input files not found.
- Errors inside the EOCFI functions.

6.2. ATTITUDE BLOCK

The Attitude Block is in charge of computing the nominal, Satellite and Restituted attitudes. The nominal attitude is the ideal attitude with no actuator errors. The Satellite attitude includes the actuator errors. The Restituted attitude includes the sensor errors.

Nominal Attitude

The Nominal Attitude is calculated from the Restituted State Vector interpolated to instrument acquisition times. The nominal attitude represents the ideal attitude with no actuation errors, and no sensor errors.

Although 6 different main attitude modes for Earth Observation missions are listed below only the Geodetic pointing and Geodetic pointing with yaw steering will be implemented in the scope of the current phase of the project:

- **Geodetic pointing.**
  Geodetic nadir pointing is when the nadir vector of the satellite is perpendicular to the local horizon. The satellite can be looking at nadir (roll angle is 0°), or it can have a roll angle.

![Figure 6-3: Geodetic and geocentric pointing attitude](image)
In imager missions it is very common to have yaw steering correction. This means that the attitude is corrected to take into account the rotation of the Earth in the sub-satellite nadir point.

![Diagram](image)

**Figure 6-4: Diagram representing the effect of Yaw Steering correction on the acquisition image.**
Left – no yaw steering. Right – with yaw steering.

**Satellite Attitude**

The Satellite attitude simulates the errors in the actuators and the platform vibrations. To do this an error function will be added to the Nominal attitude.

**Restituted Attitude**

The Restituted attitude simulates the errors in the attitude estimation by the sensors. To do this an error function will be added to the Satellite attitude. The Restituted orbit is an output in the data packages the satellite sends to Earth. It will be interpolated to the times of SC positioning system data acquisition rate.

**Attitude Error Functions**

The error functions that model the actuator and sensor errors are common for the Satellite and Restituted attitude. The errors will be configured for both. The error modelled by the error function is added to each Euler component, Yaw, Pitch Roll. Although, 4 different Error Function models are listed and described below, only Polynomial and sinusoidal Error Function will be implemented in the scope of the current phase of the project:

- **Polynomial Attitude Error Function**
  Error Function for each vector component is described by the equation:
  
  $E_{rr} = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4$

  Where:
  - $a_i$ are the polynomial coefficients, input by the user.
  - $t$ is a time in seconds since the beginning of simulation $t_0 = (MJD2000_{ini} - MJD2000) \times 86400s$
  - $E_{rr}$ is module of the vector component of the Error Function

- **Sinusoidal Attitude Error Function**
  Error Function for each vector component is described by the equation:
6.2.1. DETAILED DESCRIPTION

The following process is followed to generate Nominal, Satellite and Restituted Attitude.

Nominal Attitude simulator

1. "Attitude Mode", "Restituted Orbit", "Instrument Sampling Time" are taken as inputs (see Interfaces->Inputs, Table 6-5).

2. The Restituted orbit is interpolated to Satellite orbit times, using the EOCFI osvCompute functions. The Satellite orbit is sampled at the instrument sampling time. The Restituted orbit is sampled at the position detection sampling time (for example, the GNSS on board the spacecraft).

3. The Nominal Attitude is calculated according to the "Attitude Mode" taking into account State Vector $r_{sv}$ from the Restituted Orbit interpolated at Satellite orbit times

   a. In case of Geodetic pointing (without yaw steering) attitude mode:

      Given the state vector at an instant of time in ECI frames
      The unit vector of the position is:
      \[
      \mathbf{u}_{r_{ECI}} = \frac{\mathbf{r}_{ECI}}{||\mathbf{r}_{ECI}||}
      \]
      \[
      \mathbf{u}_{r_{ECEF}} = \frac{\mathbf{r}_{ECEF}}{||\mathbf{r}_{ECEF}||}
      \]

      The unit vector of the velocity is:
      \[
      \mathbf{u}_{v_{ECI}} = \frac{\mathbf{v}_{ECI}}{||\mathbf{v}_{ECI}||}
      \]
      \[
      \mathbf{u}_{v_{ECEF}} = \frac{\mathbf{v}_{ECEF}}{||\mathbf{v}_{ECEF}||}
      \]

      The first step is to calculate the local vertical with respect to the local horizon. This is done by:

      - Calculating the geodetic coordinates of the state vector in ECEF $r_{ECEF}^{\text{geo}}(\text{lat}, \text{lon}, h)$
      - The sub-satellite point is $r_{\text{subsc}}^{\text{geo}} = r_{ECEF}^{\text{geo}}(\text{lat}, \text{lon}, 0)$
- Calculating the ECEF state vector from the sub-satellite point
- The local vertical with respect to the local horizon is the unitary vector of the satellite to the sub-satellite point:

\[
\text{LocalVertical wrt LocalHorizon}_{\text{ECEF}} = r_{\text{SubSC}}^{\text{ECEF}} - r_{\text{SC}}^{\text{ECEF}}
\]

The local vertical is transformed to ECI frame,

\[
\text{LocalVertical wrt LocalHorizon}_{\text{ECI}} = R_{\text{ECI}}^{\text{ECEF}} \cdot \text{LocalVertical wrt LocalHorizon}_{\text{ECEF}}
\]

The Satellite Frame is defined as:

\[
\begin{align*}
Z_{\text{Sat2ECI}} &= \text{UnitVector of LocalVertical wrt LocalHorizon}_{\text{ECI}} \\
Y_{\text{Sat2ECI}} &= Z_{\text{Sat2ECI}} \times u_v^{\text{ECI}} \\
X_{\text{Sat2ECI}} &= Y_{\text{Sat2ECI}} \times Z_{\text{Sat2ECI}}
\end{align*}
\]

The satellite to ECI rotation matrix is:

\[
R_{\text{Sat}}^{\text{ECI}} = [X_{\text{Sat2ECI}} \quad Y_{\text{Sat2ECI}} \quad Z_{\text{Sat2ECI}}]
\]

The Orbital Frame is defined as:

\[
\begin{align*}
Z_{\text{Orb2ECI}} &= -u_r^{\text{ECI}} \\
Y_{\text{Orb2ECI}} &= Z_{\text{Orb2ECI}} \times u_v^{\text{ECI}} \\
X_{\text{Orb2ECI}} &= Y_{\text{Orb2ECI}} \times Z_{\text{Orb2ECI}}
\end{align*}
\]

The orbital to ECI rotation matrix is:

\[
R_{\text{Orb}}^{\text{ECI}} = [X_{\text{Orb2ECI}} \quad Y_{\text{Orb2ECI}} \quad Z_{\text{Orb2ECI}}]
\]

The ECI to Orbital rotation matrix is the inverse matrix:

\[
R_{\text{ECI}}^{\text{Orb}} = R_{\text{Orb}}^{\text{ECI}}^{-1}
\]

The attitude rotation matrix is the product of both matrices above:

\[
R_{\text{Sat}}^{\text{Orb}} = R_{\text{Orb}}^{\text{ECI}} \times R_{\text{ECI}}^{\text{Sat}}
\]

If the satellite has a user defined roll angle, the rotation matrix is translated to Euler angles, the roll added, and then translated back to rotation matrix.

b. Geodetic Pointing with Yaw Steering Correction

The reference frames are defined using the following basic vectors:

Local Nadir is a Unit vector with:

- Origin O_Y2: Centre of Mass of the complete satellite in operational conditions (in-orbit deployed configuration);
• Direction: Perpendicular to the earth’s reference ellipsoid, in the direction towards the Earth. The reference Ellipsoid used in BIBLOS is defined in the ModelId class of the EOCFs.

Velocity vector in ECEF reference frame \( \mathbf{v}_{ECEF} \) is a Vector with:

- Origin \( O_{YS} \): Centre of Mass of the complete satellite in operational conditions (in-orbit deployed configuration);
- Direction: Restituted velocity vector expressed in ECEF reference frame \( \mathbf{v}_{ECEF}^{rest} \). The Restituted velocity vector is obtained from the Restituted orbit.

Yaw Steering Reference Frame (index YS)

The Yaw-Steering Reference Frame aligns the Z-axis with the local nadir, and delivers rectangular images, provided the instrument scan is in the YZ-plane.

The Yaw Steering Reference Frame is a right-handed orthogonal frame defined by:

- Origin \( O_{YS} \): Centre of Mass of the complete satellite in operational conditions (in-orbit deployed configuration).
- \( +X_{YS} \): Roll Axis: Equal to the vector product of \( Y_{YS} \times Z_{YS} \)
- \( +Y_{YS} \): Pitch Axis: Equal to the normalised vector product of Local Nadir \( \times \) Velscearth
- \( +Z_{YS} \): Yaw Axis: Local Nadir

To apply the yaw steering the formulation is the same as for the geodetic pointing but the Satellite Frame definition is different for the Y axis.

For the yaw steering correction, to take into account the Earth rotation, the velocity vector in ECEF is transformed to ECI as if it was a vector, and not a velocity. The explanation for this is the following:

- The conversion from ECI to ECEF frame takes into account the rotations below:
  - Motion of the Celestial Pole.
  - Rotation of the Earth
  - Motion of Terrestrial Pole, including the precession and nutation.

Out of this three the dominant is the Rotation of the Earth, which is the effect that the Yaw Steering corrections

- The three rotations of the transformation from ECEF to ECI for the position vector is a matrix referred to here as \( R_{ECEF}^{ECI} \)

\[
\mathbf{r}_{ECI} = R_{ECEF}^{ECI} \cdot \mathbf{r}_{ECEF}
\]

- The velocity is the derivative of the position, so when transforming the velocity in ECI from the velocity in ECEF:

\[
\mathbf{v}_{ECI} = \frac{\partial}{\partial t} (R_{ECEF}^{ECI} \cdot \mathbf{r}_{ECEF})
\]
Calculating the velocity in ECI from the velocity in ECEF ignoring the second term (as if it was a vector, and not a velocity):

\[ \mathbf{v}_{ECI} = \mathbf{R}_{ECI}^{ECEF} \cdot \mathbf{v}_{ECEF} \]

This is what is called here the "velocity in ECEF expressed in ECI"

\[ \mathbf{v}_{(ECEF)|(ECI)} \]

The Satellite frame is defined as:

\[ \mathbf{Z}^{Sat}_{ECI} = \text{LocalVerticalwrtLocalHorizon}_{ECI} \]
\[ \mathbf{Y}^{Sat}_{ECI} = \mathbf{Z}^{Sat}_{ECI} \times \mathbf{v}_{(ECEF)|(ECI)} \]
\[ \mathbf{X}^{Sat}_{ECI} = \mathbf{Y}^{Sat}_{ECI} \times \mathbf{Z}^{Sat}_{ECI} \]

Where "\( \times \)" is the cross product operation.

In this case, with respect to the Geodetic Pointing with no Yaw Steering, the yaw steering correction is introduced in the definition of the Satellite Frame.

4. The output is **Nominal Attitude**. The Nominal orbit is sampled at Satellite orbit times.

**Satellite Attitude simulator**

5. "Attitude Actuation Error Configuration" is taken as an input.

6. Attitude Actuation Error Function is calculated for each Euler angle separately. There are two Error Function models which may be used separately or combined:

a. In case of polynomial error function:

\[ E_{rr}^j = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 \]

Where:

- \( a_i \) are polynomial coefficients, input by the user in [rad], [rad/s], [rad/s^2] etc.
- \( t \) is a time in seconds since the beginning of simulation

\[ t_0 = (MJD2000_{init} - MJD2000) \cdot 86400 \text{s} \]

- \( E_{rr}^j \) is module of the \( j \) vector component of the Error Function in [rad]

\[ E_{rr} = \begin{bmatrix} |E_{rr}^1| \\ |E_{rr}^2| \\ |E_{rr}^3| \end{bmatrix} \]
b. In case of sinusoidal error function:

\[ E_{rr}^j = a_0 \sin(2\pi a_1 t) \]

Where:

- \( a_0 \) is the amplitude of the Error Function, input by the user in [rad]
- \( t \) is a time in seconds since the beginning of simulation
- \( t_0 = (MJD2000_{init} - MJD2000) \cdot 86400 \) s
- \( E_{rr}^j \) is module of the \( j \) vector component of the Error Function in [rad]

\[ E_{rr} = \begin{bmatrix} |E_{rr}^1| \\ |E_{rr}^2| \\ |E_{rr}^3| \end{bmatrix} \]

c. In case of multiple Error Functions combined:

\[ E_{rr} = (E_{rr})_{poly} + (E_{rr})_{sin} + \ldots \]

7. Nominal attitude is converted from rotation matrix to Euler angles using transformation described in chapter §13.

8. Actuation Error Function is added to the Nominal Attitude.

\[ (R_{\text{Real}})^{\text{Euler}} = (R_{\text{Nominal}})^{\text{Euler}} + E_{rr} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} + \begin{bmatrix} |E_{rr}^1| \\ |E_{rr}^2| \\ |E_{rr}^3| \end{bmatrix} \]

9. The result is converted back to the rotation matrix form. The output is **Satellite Attitude**. The Nominal orbit is sampled at Satellite orbit times.

**Restituted Attitude simulator**

10. "Attitude Estimation Error Configuration” and "SC acquisition time” are taken as inputs.

11. Attitude Estimation Error Function is calculated for each Euler angle separately. There are two Error Function models which may be used separately or combined (analogically to Attitude Actuation Error Function, see pt.6a-6c).

12. Estimation Error Function is interpolated (spline interpolation [RD.17]) to the SC acquisition time and added to the Satellite Attitude.

13. The output is **Restituted Attitude**. The Restituted Attitude is sampled at position and orientation acquisition times (the sampling time the instrument on board the spacecraft acquire its position and orientation).

**6.2.2. FLOW DIAGRAM**

The following diagram depicts the main steps of the Attitude Building Block:
Figure 6-5 Attitude Block, Nominal, Satellite & Restituted Attitude generation
6.2.3. INTERFACES

6.2.3.1. Configuration Parameters

The following configuration parameters are defined.

- Global includes parameters that affect several modules
- Local refers to parameters that affect only this module
- Variable Type is the C data type

Table 6-4 Configuration Parameters definition for attitude block

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instrument Sampling Time</td>
<td>Time rate at which instrument samples the image</td>
<td>Double</td>
<td>s</td>
<td>Global</td>
</tr>
<tr>
<td>Attitude Mode</td>
<td>Flag describing the attitude mode to be used.</td>
<td>unsigned short</td>
<td>N.A.</td>
<td>Local</td>
</tr>
</tbody>
</table>

6.2.3.2. Inputs

Table 6-5 Attitude Block Input Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Restituted Orbit</td>
<td>Restituted orbit as calculated from the Orbit Block Size: Mx7, where M is the number of instants of time of the satellite ephemerides generation 7 elements of each row are the epoch, 3 positions and 3 velocities in GCRF</td>
<td>double</td>
<td>Earth explorer file with list of ephemeris.</td>
</tr>
</tbody>
</table>

6.2.3.3. Outputs

Table 6-6 Attitude Block Output Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal Attitude</td>
<td>Nominal attitude as calculated from the Attitude Block Size: N quaternions, where N is the number of instants of time of the instrument acquisition (Satellite orbit times)</td>
<td>double</td>
<td>Earth explorer file with list of quaternions.</td>
</tr>
<tr>
<td>Satellite Attitude</td>
<td>Satellite attitude as calculated from the Attitude Block Size: N quaternions, where N is the number of instants of time of the instrument acquisition (Satellite orbit times)</td>
<td>double</td>
<td>Earth explorer file with list of quaternions.</td>
</tr>
<tr>
<td>Restituted Attitude</td>
<td>Restituted attitude as calculated from the Attitude Block Size: N quaternions, where N is the number of instants of time of the instrument acquisition (Satellite orbit times)</td>
<td>double</td>
<td>Earth explorer file with list of quaternions.</td>
</tr>
</tbody>
</table>
6.2.4. SCOPE AND LIMITATIONS

This block is applicable to all Earth Observation missions, including missions with the following instruments:

- Passive Opticals
- Active Microwaves
- Passive Microwaves
- Active Opticals

6.2.5. EXTERNAL LIBRARIES

The libraries used inside this block are:

- Eigen, §11.1
- NetCDF §11.5
- EOCFIs §11.6
- Boost §11.4
- Logger §11.3

6.2.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfiError Error class, [RD.26]. Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception in thrown in case of error and the execution stops. Warnings do not stop the execution.

For the Attitude block the following errors can occur:

- File access: input files not found.
- Errors inside the EOCFI functions.

6.3. AOCS/INSTRUMENT COUPLING BLOCK

The AOCS/Instrument Coupling building block is in charge of simulating the pointing/attitude of the instrument so, combined with the platform AOCS, the instrument acquisition directions are determined. In addition, this building block also considers geometrical-optical properties of the instrument like Field-Of-View (FOV) or detector pitch (pixel size) and distribution.

This block obtains the viewing directions of each detector element in the instrument, reproducing the pushbroom scanning.

Two steps are considered for the AOCS/Instrument Coupling building block: LOS Simulator and Instrument Projection.

Calculation of the Line-of-Sight in Instrument Frame

The LOS calculates the detector line geometry. The output is the line-of-sight of each detector element in instrument frame. The LOS Simulator implements an ideal homothetic transformation for the geometrical model of the instrument.

Ideal Homothetic Detector geometry

The Ideal Homothetic Detector geometry model translates ideal points defined in the ideal focal plane into lines of sight. The ideal optical instrument behaviour performs a homothetic transformation between the image on ground and the image on the focal plane.

With the focal length a LOS direction (in Instrument reference frame, §13.9) to each detector element of the focal plane:
Detector misalignments and tilt errors can be simulated for the Ideal Homothetic Detector

**Instrument projection**

The Instrument Projection is in charge of reproducing the scanning and pointing of the instrument LOS as a function of time. This block couples the platform attitude with this scanning motion of the instrument. These are the platform attitude, and the instrument scanning profile in terms of an angle respect to a nominal pointing direction (nominal LOS). The nominal LOS is defined with respect to respect to the platform axis (e.g. instrument pointing in the platform –Z axis).

The thermoelastic errors that affect the instrument are included at this point. The thermoelastic models represent the errors introduced to the cameras orientation due to thermo-mechanic changes. The Instrument Projection receives the Satellite orbit and attitude from the corresponding building blocks. It calculates Satellite acquisition directions. These outputs are used by the Scene Interaction Geometry building block to obtain the Satellite and Restituted geometry-related scene data later used in the Scene Generation Module and in Level-1 Processing Module to perform the data geolocation. The scanning modes implemented are Pushbroom.

**Pushbroom**

The pushbroom scanning mode is the most extended type of scanning. In the pushbroom the detector line is perpendicular to the flight direction and all the detector elements of the detector line acquire the image at the same time.
Subpixel simulation

In order to simulate precisely the instrument behaviour, the pixels acquired for each detector element are simulated at subpixel level. For more information refer to the Instrument Module. The subpixel level is introduced in the Line-of-Sight (LOS) calculation in this Block. The LOS for each subpixel is calculated. This LOS is intersected with Earth, so the Scene produced in the Scene Generator Module is at subpixel level. In the Instrument Module, in the Spatial Block, the first step is the spatial resampling. The spatial resampling averages the subpixels into pixel level, after applying spatial aberrations, see Figure 5-2. The input to the Instrument Model is at Subpixel level, while the output to the Instrument Models is at pixel level.

In the instrument module the image is processed in units of area called Squares. This is due to memory handling limitations, as it is not possible (for average computing capabilities and operational images) to have the whole image in memory. The area to be processed is subdivided into squares, see Figure 5-1.

To simulate the Scene at subpixel level, the LOS is calculated for each subpixel. To do this each pixel is divided into the square of the scale factor subpixels. To simulate the Square Margin (the extra space area needed for adjacency purposes (see §5.4.6) in the along track direction a number of extra instrument acquisition times is simulated (the block margin divided by the scale factor). In the across track direction the detector line is artificially widened to include a number of subpixels to the side (the Square Margin).

Detector line

| | | 1 detector element = 1 pixel |

Figure 6-8: Detector line. Every instrument acquisition time, each detector element images one pixel.
Figure 6-9: Detector line simulation at subpixel level. The detector line is artificially widened a Square Margin number of subpixels

The number of subpixels imaged for every instant of time is the following:

\[ n_{\text{subpix\_act}} = n_{\text{elements}} \times SF + 2 \times SM \]

Where:
- \( n_{\text{subpix\_act}} \) = number of subpixels imaged across track
- \( n_{\text{elements}} \) is the number of detector elements
- \( SF \) is the Scale Factor
- \( SM \) is the Square Margin

\[ n_{\text{subpix\_alt}} = SF \]

Where:
- \( n_{\text{subpix\_alt}} \) = number of subpixels imaged along track
- \( SF \) is the Scale Factor

\[ n_{\text{subpix}} = n_{\text{subpix\_act}} \times n_{\text{subpix\_alt}} \]

Where:
- \( n_{\text{subpix}} \) = number of subpixels imaged every instrument acquisition time
- \( n_{\text{subpix\_alt}} \) = number of subpixels imaged along track
- \( n_{\text{subpix\_act}} \) = number of subpixels imaged across track

6.3.1. DETAILED DEFINITION

The following process is followed to generate the Line-of-sight (LOS) in instrument and ECEF frames.

Calculation of the LOS in Instrument Frame

7. The first step is to calculate the position of each of the detector elements in the focal plane.
   1. The ideal LOS assumes that the detector line is straight. The position \((x,y)\) of each subpixel element is calculated with a straight line that includes the tilt (slight rotation of the detector line) and the displacement (shift in position in the focal plane).
2. The LOS is calculated as a homothetic transformation with the Antenna Focal Length (z coordinate). The unitary vector of the LOS is in the direction of (x,y,z).

Calculation of the position of each subpixel element in the focal plane (x, y).

The detector line central position is a configuration parameter \((centre_x, centre_y)\), as well as the number of detector elements \((n\_pix)\), the size of each detector element \((pix\_size)\), the scale factor \((SF)\) and the Square Margin \((SM)\).

a. Calculation of the number of subpixels across track (direction x).

Each pixel is divided into the square of the scale factor subpixels (for example is SF=3, there are 9 subpixels simulated per pixel). Additionally there is the Square Margin (see §5.4.6).

Along track the number of subpixels is equal to the scale factor.

Across track the number of subpixels is the scale factor times the number of pixels plus two margins (one on each side). See Figure 6-9

\[
 n\_subpix\_act = n\_elements \times SF + 2 \times SM
\]

Where:
- \(n\_subpix\_act\) = number of subpixels imaged across track
- \(n\_elements\) is the number of detector elements
- SF is the Scale Factor
- SM is the Square Margin

b. Size of each subpixel

The size of each subpixel is the size of the pixel divided by the scale factor.

\[
 subpix\_size = pix\_size/SF
\]

c. Calculation of the position of the first subpixel in across track (direction x)

The position of the so-called first subpixel is calculated with the central position of the detector line provided by the user \((centre_x)\), the number of pixels across track \((n\_subpix\_across)\), calculated before, and the size of each subpixel \((subpix\_size)\).

\[
 First\ subpixel \quad (X<0; \ Y>0)
\]

\[
\begin{array}{c}
1,1 \\
1,2 \\
2,1 \\
\end{array}
\]

\[
 Y\_inst \quad X\_inst
\]

Figure 6-10: Location of the first subpixel. The first subpixel is in the Square Margin, and has X<0 and Y>0 coordinates.
If the number of subpixels across track is ODD the centre position corresponds to the centre of the central subpixel.

\[
\text{first\_subpix\_x} = \text{centre\_x} - \frac{(n_{\text{subpix\_across}} - 1)}{2} \times \text{subpix\_size}
\]

If the number of subpixels across track is EVEN the centre position is in the border of two subpixels.

The same formula that above applies, see the figures below.

- When the number of subpixels across track is ODD:

\[
\Delta y \quad \Delta x
\]

\[
\text{Centre position of the detector line (x,y)}
\]

**Figure 6-11:** Location of the centre position of the detector line when the number of subpixels across track is ODD, is in the centre of a subpixel.

- When the number of subpixels across track is EVEN:

\[
\Delta y \quad \Delta x
\]

\[
\text{Centre position of the detector line (x,y)}
\]

**Figure 6-12:** Location of the centre position of the detector line when the number of subpixels across track is EVEN, is between two subpixels.

d. Calculation of the position of the first subpixel in along track (direction y)

The number of subpixels in the along track direction is the scale factor. A similar issue than for the across track calculation, if the scale factor is odd the centre of the detector line is also the centre of a subpixel. If the scale factor is even the centre of the detector line is in the border of two subpixels. The calculation is the following.
first_subpix_y = centre_y + (SF-1)/2*subpix_size

e. Calculation of the position across and along track of each subpixel (coordinate x)
The position across track of each subpixel is the following
pos_x(i,j) = first_subpix_x + (j-1)*size_subpix
pos_y(i,j) = first_subpix_y - (i-1)*size_subpix

Note that the X coordinate of the subpixels increases with respect to the first subpixel, while the Y coordinate of the subpixels decreases.

Figure 6-13: Position of a subpixel (i,j) w.r.t. the position of the first subpixel.

Inversion of the Focal Plane

The Focal Plane can be inverted or not depending on the optics of the instrument. To avoid confusion, in BIBLOS the Focal Plane will not be inverted. If the instrument has an inverted focal plane, the user can easily adapt it by multiplying the focal plane (x,y) coordinates by -1.

Figure 6-14: Generation of LOS direction (in camera frame), knowing location at the focal plane and focal length
8. The output is the **LOS in Instrument frame** for all subpixel elements.

**Projection of the LOS in ECEF Frame**

Once the LOS in Instrument Frame is calculated, it is necessary to transform it to ECEF in order to intersect with the ellipsoid and calculate where each detector element is looking at for each instant of time. In order to do this it is necessary to obtain the Instrument to ECEF rotation matrix for every instant of acquisition time.

1. The first step is to calculate the rotation matrix between the instrument and the Satellite, $\text{RotMat}_\text{Inst2Sat}$. This is the definition of how the instrument is mounted on the platform. The user inputs the Euler angles that define the rotation of the instrument with respect to the platform, keeping in mind that a 3-2-1 convention is applied. The reference direction is nadir looking, the vector in Satellite frame is $[0,0,1]$. If there are no rotations applied ($\text{RotMat}_\text{Inst2Sat}$ is the identity matrix), the LOS in instrument frame is looking at nadir.

The user defined instrument to satellite rotation is applied:

$$\text{RotMat}_\text{Inst2Sat} = \text{RotMat}_\text{roll} \times \text{RotMat}_\text{pitch} \times \text{RotMat}_\text{yaw}$$

Where $\text{RotMat}_\text{roll}$, $\text{RotMat}_\text{pitch}$ and $\text{RotMat}_\text{yaw}$ are the unitary rotation matrices defined in [PD.2].

If the satellite is looking with a look angle of $30^\circ$, the instrument_to_satellite_rotation will be $[30\pi/180,0,0]$.

Note that a rotation of $+30^\circ$ in the $+X$ axis, i.e. left-side looking.
A rotation of -30° is negative around the X axis and would be to the right-side of the instrument in the direction of the velocity.

2. **Thermoelastic error modelling** is introduced in the Instrument to Satellite rotation matrix. These errors represent the thermo-mechanic variations in the detector lines that happen at different stages of the orbit. The period of these errors is around one orbit, and it is much larger than the typical acquisition image. Due to this, the thermoelastic errors are taken as a constant during the acquisition of the image. To define the error, a roll, pitch and yaw angle are provided via configuration file.

The thermoelastic errors are pre-multiplied to the instrument to rotation matrix.

\[
\text{RotMat}_{\text{Inst2Sat}} = \text{RotMat}_{\text{ThermoelasticErr}} \times \text{RotMat}_{\text{Inst2Sat}};
\]

3. The next step is to calculate the rotation matrix between the orbital frame and ECEF, \(\text{RotMat}_{\text{Orb2ECEF}}\). This transformation is defined in §13.10[PD.2]. The orbital to ECEF frame is calculated for the Satellite and Restituted orbit times (instrument acquisition time, and ephemerides generation time).

4. To calculate the Instrument to ECEF conversion, the rotation matrices are multiplied:

\[
\text{RotMat}_{\text{Instr2ECEF}} = \text{RotMat}_{\text{Orb2ECEF}} \times \text{Attitude} \times \text{RotMat}_{\text{Instr2Sat}}
\]

The instrument to ECEF frame is calculated for the Satellite and Restituted orbit times (instrument acquisition time, and ephemerides generation time).

5. The LOS in ECEF is calculated by multiplying the LOS in instrument frame by the rotation matrix:

\[
\text{LOS}_{ECEF} = \text{RotMat}_{\text{Instr2ECEF}} \times \text{LOS}_{\text{Instr}}
\]

The LOS in ECEF is a function of time. Only the first and last position are saved to file (LOS_ECEF_ini, LOS_ECEF_end). In the scene interaction geometry, when the LOS is intersected with the ellipsoid, the LOS will be calculated for each instant of time within the loop.

6.3.2. **FLOW DIAGRAM**

The following diagram depicts the main steps of the orbital Building Block.
Figure 6-16 Calculation of the LOS in instrument frame
6.3.3. INTERFACES DEFINITION

In these sections the interfaces shall be defined including Configurations Parameters, Inputs and Outputs.

Figure 6-17 Projection of the LOS in ECEF
6.3.3.1. Configuration Parameters

The following configuration parameters are defined.

- Global includes parameters that affect several modules
- Local refers to parameters that affect only this module
- Variable Type is the C data type, from [RD.10]

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermoelastic_errors</td>
<td>Thermoelastic errors. Three values for [roll, pitch, yaw]. The values are taken as constant during the acquisition (see section 2 above for justification)</td>
<td>Double</td>
<td>radians</td>
<td>Local</td>
</tr>
<tr>
<td>Instrument_to_satellite_rotation</td>
<td>The mounting of the instrument in the platform is defined here via the three Euler angles (that will be applied 3-2-1). Three values for [roll, pitch, yaw]. The Euler angles will define the position of the line-of-sight in the Satellite frame. If the angles are 0s, the instrument will be looking with a reference direction of [0,0,1], i.e., to Nadir (see definition of the Satellite Frame in the Technical Specifications document). If the satellite is looking with a look angle of 30°, the instrument_to_satellite_rotation will be [30°pi/180,0,0]. Note that a rotation of +30° in the +X axis, i.e. left-side looking. A rotation of -30° is negative around the X axis and would be to the right-side of the instrument in the direction of the velocity.</td>
<td>Double</td>
<td>radians</td>
<td>Global</td>
</tr>
<tr>
<td>Number_of_pixels</td>
<td>Number of pixels in the detector line</td>
<td>Unsigned int</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Pixel_size</td>
<td>Size of each detector element, which is assumed to be a perfect square.</td>
<td>Double</td>
<td>m</td>
<td>Global</td>
</tr>
<tr>
<td>Focal_plane_position</td>
<td>Position of the focal plane centre in instrument frame (x,y,z)</td>
<td>Double</td>
<td>m</td>
<td>Global</td>
</tr>
<tr>
<td>Effective_focal_length</td>
<td>Antenna focal length of the telescope</td>
<td>Double</td>
<td>m</td>
<td>Global</td>
</tr>
<tr>
<td>Detector_displacements</td>
<td>Focal plane position error in instrument frame (x,y)</td>
<td>Double</td>
<td>Meters</td>
<td>Local</td>
</tr>
<tr>
<td>Detector_missalignments</td>
<td>Focal plane error in rotation from the first detector element.</td>
<td>Double</td>
<td>Radians</td>
<td>Local</td>
</tr>
<tr>
<td>Scale factor</td>
<td>The pixel shall be divided into a number of pixels equal to the square of the scale factor. If the resampling factor is 3 for example, there are 9 subpixels per pixel imaged.</td>
<td>Unsigned int</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Variable Type</td>
<td>Units</td>
<td>Global or Local</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------</td>
<td>---------------</td>
<td>----------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Square margin</td>
<td>Number of Subpixels that are simulated in the processing of each square in the spatial block. Each area of image processed shall have an excess margin of image that will be cut in the spatial block. This area is needed for adjacency calculations and for border effects. The square margin is related to the scale factor, input square size (to the spatial block) and output square size (to the spatial block), with this relation (Input square size - 2* square margin)/(scale factor) = output square size For example, if scale factor = 3; square margin = 62; input square size = 1024 and input square size = 300. The following relationship is met: (1024 - 2*62)/3 = 300</td>
<td>Unsigned integer</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Input square size</td>
<td>INPUT Size of the square that are processed in the spatial block. It is only used in this block to verify that the relation between the input square size, output square size, square margin and scale factor is met.</td>
<td>Unsigned integer</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Output square size</td>
<td>OUTPUT Size of the square that are processed in the spatial block. It is only used in this block to verify that the relation between the input square size, output square size, square margin and scale factor is met.</td>
<td>Unsigned integer</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
</tbody>
</table>

**6.3.3.2. Inputs**

**Table 6-8 Inputs definition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellite orbit</td>
<td>Satellite orbit as calculated from the Orbit Block</td>
<td>Earth explorer file with list of state vectors.</td>
<td>[mjd2000, m,m,m, m/s,m/s,m/s]</td>
</tr>
<tr>
<td>Satellite attitude</td>
<td>Satellite attitude as calculated from the Attitude Block</td>
<td>Earth explorer file with list of state vectors.</td>
<td>Dimensionless</td>
</tr>
</tbody>
</table>

**6.3.3.3. Outputs**

**Table 6-9 Outputs definition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOS_Instrument</td>
<td>Line-of-Sight in instrument direction. Unitary vector with the dimension of Nx3, where N is the number of detector elements. The columns are (x,y,z), where z is the antenna focal length, and (x,y) is the position on the focal plane. The LOS is divided by the modulus so that it is a dimensionless vector.</td>
<td>Ascii file with list of vectors</td>
<td>Dimensionless</td>
</tr>
</tbody>
</table>
6.3.4. SCOPE AND LIMITATIONS

This block is applicable to all Earth Observation missions, including missions with the following instruments:

- Passive Opticals
- Active Microwaves
- Passive Microwaves
- Active Opticals

6.3.5. EXTERNAL LIBRARIES

The libraries used inside this block are:

- Eigen, §11.1
- NetCDF §11.5
- EOCFIs §11.6
- Boost §11.4
- Logger §11.3

6.3.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfError Error class, [RD.26].

Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception in thrown in case of error and the execution stops. Warnings do not stop the execution.

For the AOCS/Instrument Coupling block the following errors can occur:

- File access: input files not found.
- Errors inside the EOCFI functions.

6.4. SCENE INTERACTION BLOCK

The Scene Interaction Geometry Block projects the pixel array onto the Digital Elevation Model (DEM). It is catered for Earth observing missions only. Atmospheric and calibration scenes are not taken into account in this block.

The inputs to this block are the line-of-sight (LOS) in Instrument frame, the rotation matrix from Instrument to ECEF frame, calculated in the AOCS/Instrument Coupling Block, the Satellite orbit and the DEM. The output of this block is the observation area in latitude and longitude.

This block additionally computes, based on the Satellite orbit and attitude, the additional scene data needed for scene generation like Solar Zenith Angle (SZA), Observation Zenith Angle (OZA) and the Relative Azimuth Angle (RAA). These angles are inputs to the Scene Generator Module.
Intersection with DEM

In the case of use of DEM, the terrain elevation is taken into account and intersected with the LOS. In Figure 6-18 the diagram of acquisition with DEM is represented. If no DEM is taken into account in the processing, the LOS acquires the centre point of the reference image (in the example of the figure). If the DEM is taken into account, and a mountain is located in the reference image, the LOS intersects with the mountain and the georeferenciated position in the reference image is not the centre of the image.

To calculate the intersection of the LOS with the DEM, an iterative method is followed detailed here and described in Figure 6-20. The use of the DEM is done through the EOCFI libraries. The initial value is taken from the value of the previous pixel, and the initial value of the simulation is taken as 0 meters of altitude. An illustrative value, the number of iterations is typically 2-3 for a precision in altitude of around 0.1m for terrains with no radical steep changes. In terrains with steep changes in altitude, like the Himalayas, for example, the loop can diverge, so via configuration file the maximum number of iterations is configured. When the no convergence value is reached, the altitude assigned to the pixel is the intersection with the Ellipsoid. This will be used in the access to the Atmospheric look up tables.
The algorithm stops when the geodetic distance between the computed intersection $M_i$ and the one computed at the previous step becomes below a fixed threshold ($d_{min}$).

The visibility of the scene using a DEM produces several situations depending on the terrain orography. These cases are represented in Figure 6-21, Figure 6-22, Figure 6-23 and Figure 6-24.

For a flat orography, flat meaning that the DEM heights applied to the ellipsoid are 0 for the entire reference image, the acquisition of the image is complete, as represented in Figure 6-21.
If there is a mountain and the reference image is located so that the slope is seen as ascending with respect to the acquisition of the satellite, the acquisition beam cuts the slope and images only part of the scene, see Figure 6-22, resulting in a partial acquisition of the reference image.

On the other hand, if there is a presence of a hill with a descending slope with respect to the acquisition of the satellite, with a slope lower that the acquisition’s beam tangent, this results in an imaged terrain bigger than the reference image, as seen in Figure 6-23.
The last possibility is that there is a mountain with a steep descending slope in the reference image, steeper than the acquisition beam’s tangent, see Figure 6-24, so that there is an area from which the acquisition beam is not receiving information, named “shadow area” in the figure. This must be taken into account because a part of the reference image will be left un-imaged.

Additionally, a combination of the above can take place in the same reference image, especially in mountainous areas or areas with a changing orography.

Definition of the OZA, SZA, RAA
The Observation Zenith Angle (OZA) is the angle between the satellite to observation point direction and the azimuth of the observation point.

The Sun Zenith Angle (OZA) is the angle between the sun to observation point direction and the azimuth of the observation point.

The OZA and SZA, are shown in the following diagram:

![Diagram showing OZA and SZA](image)

**Figure 6-25: Observation zenith angle, Sun zenith angle**

The Relative Azimuth Angle (RAA) is the angle between the ground projection of the Sun-observation point and the Satellite-observation point directions:
Figure 6-26: Relative Azimuth Angle

Limb sounding missions.

The current implementation of the Scene Interaction does not cover another type of missions: the limb-sounding missions. In this case the satellite is pointing with a direction close to the horizon. These missions usually aim measure Atmospheric properties. An example is the CEOS Atmospheric Chemistry Virtual Constellation.
6.4.1. DETAILED DEFINITION

The Scene Generation Block has a double loop:
- First for each subpixel in the along track direction.
- Then for each subpixel in the across track direction.

The output of this block is the acquisition area, the scene, what the detector line is looking at. The information provided is the geodetic coordinates (lat,lon,h), and the geometrical angles (sun zenith angle, observation zenith angle and relative azimuth angle).

This block’s outputs are heavy (binary matrices of doubles) for each position and detector element. As the image can be quite large scene is divided into smaller parts to manage it in memory. A configurable number of rows (acquisition instants) is processed and written to file. The number of columns is always the number of subpixels (for calculation of the number of subpixels see §6.3).

![Figure 6-27: Typical pointing for a limb-sounding mission. Source: EOCFI Pointing Library documentation, [RD.26]](image)

![Figure 6-28: Scene Interaction Block memory management](image)
Before the loop the Scene interaction does the following:

1. Load Satellite orbit in ECEF
2. Load instrument to ECEF quaternions
3. Load the Line of Sight in Instrument Frame
4. Calculates the optical centre displacement in ECEF (distance between the instrument and the centre of mass of the satellite)
5. Calculate the Sun ephemeris, with EOCFI (see §9.), setSun of the class State Vector.
6. Reserve memory for the outputs (geodetic coordinates and observation angles)
   - Geodetic matrix of N_lines x n_subpix_act x 3 [latitude, longitude, altitude]
   - Sun zenith angle matrix of N_lines x n_subpix_act
   - Observation zenith angle matrix of N_lines x n_subpix_act
   - Relative azimuth angle matrix of N_lines x n_subpix_act

Where N_lines is the scale factor times the output square size and

And:

\[ \text{n_subpix_act} = \text{n_elements} \times \text{SF} + 2 \times \text{SM}. \text{See Figure 6-9} \]

Where:
- \text{n_subpix_act} = number of subpixels imaged across track
- \text{n_elements} is the number of detector elements
- \text{SF} is the Scale Factor
- \text{SM} is the Square Margin

The steps inside the loop are the following:

1. Calculation of the Line-of-Sight (LOS) in ECEF with the LOS in Instrument frame and the rotation matrix (which is calculated from the corresponding quaternions read from file using the EOCFI libraries).
   \[ \text{LOS}_{\text{ECEF}} = \text{RotMat}_{\text{Instr2ECEF}} \times \text{LOS}_{\text{Instrument}} \]

2. Calculation of the intersection with the Ellipsoid or with the DEM
   a. If the user has selected to not use the DEM (Flag_use_DEM=0), the intersection is done with the Ellipsoid. In this case the intersection is a direct geometrical calculation.

**Intersection with the Ellipsoid**

The point of application of the Line-of-Sight is the orbit position plus the translation of the detector line in ECEF:

\[ \text{pixPOS}_{\text{ECEF}} = \text{translation}_{\text{ECEF}} + \text{orbit}_{\text{ECEF}} \]

If \([X_p, Y_p, Z_p]\) are the coordinates of \text{pixPOS}_{\text{ECEF}} and \([l,m,n]\) the direction cosines of the Line-of-sight in ECEF

Equatorial an Polar radius of the ellipsoid plus a height above it, and \text{a_wgs} is the equatorial radius of the Ellipsoid.

\[ \text{Re} = \text{a_wgs} + \text{height}; \]
\[ \text{Rp} = (\text{a_wgs} \times (1 - \text{flattening})) + \text{height}; \]

The system of 2 equations is solved:

The point is on the Earth’s surface: \((X/\text{Re})^2 + (Y/\text{Re})^2 + (Z/\text{Rp})^2 = 1\)

The line of intersection is parametrised with \([X_p, Y_p, Z_p] + t*[l,m,n]\)
The solution for \( t \) is:

\[
\begin{align*}
t &= \frac{-1}{(R_p^2 (l^2 + m^2) + R_e^2 n^2)} \times \left( R_p^2 (l X_p + m Y_p) + R_e^2 n Z_p + \frac{1}{2} \sqrt{4 (R_p^2 (l X_p + m Y_p) + R_e^2 n Z_p)^2 - 4 (R_p^2 (l^2 + m^2) + R_e^2 n^2) (R_p^2 (-R_e^2 + X_p^2 + Y_p^2) + R_e^2 Z_p^2))} \right)
\end{align*}
\]

Identifying coefficients for the quadratic equation, with the following coefficients:

\[
\begin{align*}
A &= (l^2 + m^2)(R_p^2) + (n^2)(R_e^2); \\
B &= 2.0 \times \left( (l X_p + m Y_p)(R_p^2) + n Z_p(R_e^2) \right); \\
C &= (X_p^2 + Y_p^2)(R_p^2) + (Z_p^2)(R_e^2) - (R_e^2)(R_p^2);
\end{align*}
\]

Solution of the quadratic equation (the smaller of the two):

\[
\alpha = \frac{-1}{2} \times \left( B + \sqrt{B^2 - 4AC} \right) / (2A);
\]

ECEF coordinates of the intersection with Earth’s surface are \([X,Y,Z]\):

\[
\begin{align*}
X &= X_p + (l \alpha) \\
Y &= Y_p + (m \alpha) \\
Z &= Z_p + (n \alpha)
\end{align*}
\]

b. If the user has selected to use the DEM (Flag_use_DEM=1), the intersection is done with an iterative method until convergence is reached, see Figure 6-20.

**Intersection with the Digital Elevation Model (DEM)**

To calculate the intersection with the DEM there will be an iteration loop that will be exited when a convergence threshold is met.

When convergence is not met, an internal iterator is increased. The Logger will print the percentage of pixels that have not converged. There is no error message when convergence is not met, because if the input DEM threshold is too strict and the maximum number of iterations low, the non-convergence will occur for every single pixel, and this will penalise the performances due to calls to the Logger. The percentage will give the information without collapsing the execution.

First, calculate the intersection with the Ellipsoid, \( \text{pointEarth} \)

\[
\begin{align*}
Re &= a_{wgs} + \text{height}; \\
Rp &= (a_{wgs} \times (1 - \text{flattening})) + \text{height}; \\
\text{pointEarth} &= \text{calculateIntersectionWithEarthEllipsoid(pixPOS_ECEF, LOS_ECEF)};
\end{align*}
\]

The \text{calculateIntersectionWithEarthEllipsoid} function is the BIBLOS function defined in the section above.

Second, calculate the geodetic coordinates of \( \text{pointEarth}[X,Y,Z] \), \([\text{latitude}, \text{longitude}, \text{height}]\).

Third, Retrieve the height from the DEM

\[
\text{new_height} = \text{getHeight(latitude, longitude)};
\]

The handling to the DEM is done via the EOCFI s. The getHeight is an EOCFI function (from the DemId class, the compute function is used).

Fourth, calculate the ECEF coordinates with the new height found by the DEM \([\text{latitude}, \text{longitude}, \text{new_height}]\). The new ECEF coordinates are \([X_{\text{new}}, Y_{\text{new}}, Z_{\text{new}}]\)

Finally, calculate the distance between the position from the previous and the current loop

\[
\text{difference_position} = \sqrt{(X_{\text{new}} - X_{\text{previous}})^2 + (Y_{\text{new}} - Y_{\text{previous}})^2 + (Z_{\text{new}} - Z_{\text{previous}})^2};
\]
When this difference in position is smaller than the threshold, the iteration has converged and the intersection is in \([X_{new}, Y_{new}, Z_{new}]\). The corresponding geodetic coordinates are stored.

The loop has a break condition with a maximum number of iterations in case convergence is not met. When convergence is not met, a non-convergence counter is increased (printing in log can collapse the log file, so a counter is used to count the number of pixels that do not reach convergence). The value taken with non-convergence is a direct intersection with the Ellipsoid plus the altitude of the last pixel. Non-convergence can happen due to multiple reasons. A frequent reason can be that the user inputs a small tolerance, for example 0.001m, with a small number of maximum iterations, for example 2-3. The altitude of the DEM is used in the Scene Generation module to access the Look-up-tables. If there is not convergence, the simulation can still be run, and the LUTs will be accessed with altitude 0.0, which can still be interesting as a result if the simulation is not aimed at precise radiance simulation (for example for simulations testing the Geometry errors, the instrument errors, the calibration etc). Therefore a message is printed in the Log but no exception is thrown, so the user can decide whether to discard or not the products.

The DEM is read with the EOCFI library DemId, see §9.

3. Calculation of the Observation Zenith Angle (OZA),
The position of the pixel on ground in ECEF is \(\text{PixelPositionECEF}\) and the orbit position in ECEF is \(\text{orbitECEF}\).

The vector from the pixel to the satellite
\[
\text{FromPixelToSat} = \text{orbitECEF} - \text{pixPOS\_ECEF};
\]
\(\text{UnitaryPixelToSat}\) unitaryVector(FromPixelToSat);

Unitary vectors are done through the Eigen library.

Calculation of the local vertical. First calculation of the subsatellite point by calculating the ECEF coordinates from the geodetic coordinates with altitude zero. The geodetic coordinates of the position have been calculated in the section above.
\[
[\text{PixelX}, \text{PixelY}, \text{PixelZ}] = \text{GeodeticToCartesian}(\text{PixelLat}, \text{PixelLon}, 0.0);
\]

\text{GeodeticToCartesian} is an EOCFI function. All reference changes in BIBLOS are done through the EOCFI library.

To calculate the local vertical the point at 1 meter of altitude is calculated. 1 meter is a random value higher than 0. To calculate the local horizon the Cartesian coordinates of two points with the same geodetic latitude and longitude, but with different altitudes is taken and subtracted.
\[
[\text{PixelX\_aux}, \text{PixelY\_aux}, \text{PixelZ\_aux}] = \text{GeodeticToCartesian}(\text{PixelLat}, \text{PixelLon}, 1.0);
\]

\(\text{FromPixelToLocalVertical} [0] = \text{PixelX\_aux} - \text{PixelX};\)
\(\text{FromPixelToLocalVertical} [1] = \text{PixelY\_aux} - \text{PixelY};\)
\(\text{FromPixelToLocalVertical} [2] = \text{PixelZ\_aux} - \text{PixelZ};\)
\(\text{UnitaryNormal}\) unitaryVector(FromPixelToLocalVertical);

The OZA is the angle between the vector of the pixel to the satellite and the local vertical.
\(\text{OZA} = \text{acos}(\text{dot}(\text{UnitaryNormal}, \text{UnitaryPixelToSat}));\)

4. Calculation of the Sun Zenith Angle (SZA)
The calculation of the SZA is very similar but with the angle between the pixel and the Sun and the local vertical.
\(\text{FromPixelToSun} = \text{SunPositionECEF} - \text{PixelPositionECEF};\)
UnitaryPixelToSun.unitaryVector(FromPixelToSun);
SZA = acos(dot(UnitaryNormal, UnitaryPixelToSun))

The position of the Sun in ECEF is obtained through the EOCFIs, with the StateVector class.

5. Calculation of the Relative Azimuth Angle (RAA)

Calculation of the vectors that define the RAA, see Figure 6-26. First calculation of the projection of the sun to pixel vector on ground.

CrossTangent.cross(UnitaryNormal, UnitaryPixelToSun);
UnitaryCrossTangent.unitaryVector(CrossTangent);
ThirdVector.cross(UnitaryCrossTangent, UnitaryNormal);
UnitaryThirdVector.unitaryVector(ThirdVector);

Projection of the Satellite to pixel on ground:
InLVLHPlane = UnitaryPixelToSat - (UnitaryNormal) * cos(OZA);
UnitInLVLHPlane.unitaryVector(InLVLHPlane);

RAA is the angle between those two vectors:
RAA = acos(dot(UnitaryThirdVector, UnitInLVLHPlane))
Open output files.
Reserve memory for the output variables

for iorb=1:nOrb
  Initialise variables to zeros
  for irow=A:B (A and B are the start and end orbit positions of the number of rows that are kept in memory before writing to file)
    for iele=1:nElements
      Intersection with the DEM (or Ellipsoid) and calculation of the SZA, OZA, RAA
    end
  end
end

Append the lat/ion/alt/SZA/OZA/RAA to their corresponding output files

Close output files.

Figure 6-29: Scene Interaction Block architecture and memory management
Inside the loop, the calculations for each detector element and time are shown in this figure:
6.4.3. INTERFACES DEFINITION

In this sections the interfaces shall be defined including Configurations Parameters, Inputs and Outputs.

6.4.3.1. Configuration Parameters

The following configuration parameters are defined.

- Global includes parameters that affect several modules
- Local refers to parameters that affect only this module
- Variable Type is the C data type, from [RD.10]

The Ellipsoid model used in BIBLOS correspond to the default model of the ModelId class of the EOCFIs, which is:

- Flattening \([-\cdot]\): 0.00335281
- Major Axis \([m]\): 6.37814e+06
- Eccentricity \([-\cdot]\): 0.0818192

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene_geo_filename</td>
<td>Name of the output geodetic (latitude, longitude, altitude) filename</td>
<td>Char</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Scene_SZA_filename</td>
<td>Name of the output SZA filename</td>
<td>Char</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Scene_OZA_filename</td>
<td>Name of the output OZA filename</td>
<td>Char</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Scene_RAA_filename</td>
<td>Name of the output RAA filename</td>
<td>Char</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Displacement_of_focal_plane_wrt_Centre_of_mass</td>
<td>The focal plane is not positioned in the centre of mass, where the orbit position is located. This coordinates are the displacement of the focal plane with respect to the centre of mass in Instrument Frame.</td>
<td>Double</td>
<td>m</td>
<td>Local</td>
</tr>
<tr>
<td>Flag_use_DEM</td>
<td>Flag to select whether to use the DEM or not. If Flag_use_DEM=0 no DEM is used, and the intersection is done with the ellipsoid If Flag_use_DEM= the DEM is used.</td>
<td>boolean</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>DEM_config_file</td>
<td>This file is read by the EOCF DemId Class. There is information of the DEM, the folder where it is stored, etc.</td>
<td>Char</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>DEM_convergence_threshold</td>
<td>Threshold of convergence for the DEM. Lowering the threshold can increase the performances exponentially, and DEMs carry a degree of error, so unless high precision is needed a value of 0.1 is recommended.</td>
<td>Double</td>
<td>Meters</td>
<td>Global</td>
</tr>
<tr>
<td>DEM_max_number_iterations</td>
<td>Maximum number of iterations for the DEM. Convergence is usually reached in 2-3 iterations for a threshold of 0.1 meters for areas with gradual landscape (not steep mountains).</td>
<td>Unsigned int</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
</tbody>
</table>
6.4.3.2. Inputs

Table 6-11 Inputs definition

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellite orbit</td>
<td>Satellite orbit as calculated from the Orbit Block. Size: Nx7, where N is the number of instants of</td>
<td>Earth explorer file with list of</td>
<td>[mjd2000, m,m,m, m/s,m/s,m/s]</td>
</tr>
<tr>
<td></td>
<td>time of the instrument acquisition</td>
<td>state vectors.</td>
<td></td>
</tr>
<tr>
<td>LOS_Instrument</td>
<td>Line-of-Sight in instrument direction. Unitary vector with the dimension of Nx3, where N is the number</td>
<td>Ascii file with list of vectors</td>
<td>Dimensionless</td>
</tr>
<tr>
<td></td>
<td>of detector elements. The columns are (x,y,z), where z is the antenna focal length, and (x,y) is the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>position on the focal plane. The LOS is divided by the modulus so that it is a dimensionless vector.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quat_Instr2ECEF</td>
<td>Quaternions from Instrument to ECEF frame for each instant of acquisition time. Size: N quaternions,</td>
<td>Earth explorer file with list of</td>
<td>Dimensionless</td>
</tr>
<tr>
<td></td>
<td>where N is the number of instants of time of the instrument acquisition (Satellite orbit times)</td>
<td>state vectors.</td>
<td></td>
</tr>
<tr>
<td>DEM</td>
<td>Digital Elevation Model</td>
<td>ACE2 files.</td>
<td>m</td>
</tr>
</tbody>
</table>

6.4.3.3. Outputs

Table 6-12 Outputs definition

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene_geodetic</td>
<td>Output Acquisition scene matrix of size NxMx3. If the pass is descending the top row is the northern</td>
<td>NetCDF file, double values saved in</td>
<td>rad, rad, m</td>
</tr>
<tr>
<td></td>
<td>part of the image, if the pass is ascending the first row is the southernmost point. The first</td>
<td>double precision</td>
<td></td>
</tr>
<tr>
<td></td>
<td>detector element shall be the first defined in the input LOS_instrument file. This matrix contains</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the latitude, longitude and altitude geodetic coordinate. N is the number of subpixels along track.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>M is the number of subpixels across track</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scene_SZA</td>
<td>This matrix contains the Sun Zenith Angle. The size is NxMx1.</td>
<td>NetCDF file, double values saved in</td>
<td>rad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>double precision</td>
<td></td>
</tr>
<tr>
<td>Scene_OZA</td>
<td>This matrix contains the Observation Zenith Angle. The size is NxMx1</td>
<td>NetCDF file, double values saved in</td>
<td>rad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>double precision</td>
<td></td>
</tr>
<tr>
<td>Scene_RAA</td>
<td>This matrix contains the Relative Azimuth Angle. The size is NxMx1</td>
<td>NetCDF file, double values saved in</td>
<td>rad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>double precision</td>
<td></td>
</tr>
</tbody>
</table>
6.4.4. SCOPE AND LIMITATIONS

This block is applicable to all Earth Observation missions, including missions with the following instruments:

- Passive Opticals
- Active Microwaves
- Passive Microwaves
- Active Opticals

The limitation is that this block is for Earth observing missions only. Atmospheric and calibration scenes are not taken into account in this block. The Line-of-sight is intersected with the Earth’s ellipsoid, so scenes that are not on the Earth are out of scope for this block.

6.4.5. EXTERNAL LIBRARIES

The libraries used inside this block are:

- Eigen, §11.1
- NetCDF §11.5
- EOCFs §11.6
- Boost §11.4
- Logger §11.3

6.4.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfiError Error class, [RD.26]. Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception in thrown in case of error and the execution stops. Warnings do not stop the execution.

For the Scene Interaction block the following errors can occur:

- File access: input files not found.
- Errors inside the EOCFI functions.
- The Line-of-Sight does not intersect with the Ellipsoid. The current version of BIBLOS is designed for Earth observing scenes, so if there is no intersection with the Ellipsoid an exception is thrown. Information of the position of the pixel in ECEF, the Line of Sight and auxiliary information is printed on screen so that the user can track the error, which is probably is the configuration (the orientation of the instrument, focal plane, or attitude).
7. SCENE GENERATION MODULE

The Scene generation module architecture is in the following figure. It is composed of two blocks, the resampling, and the Atmospheric blocks.

![Figure 7-1: Scene Generation Module](image)

7.1. RESAMPLING BLOCK

The Resampling block interpolates the input external image from the Scene Generator Module to the observation area as calculated by the Geometry Module. The external input image to this block is either an image in Top-of-Canopy (TOC) reflectances or an image in the biophysical parameter. The output of the block is the same (TOC or bio/geophysical parameter), resampled to the lat/lon grid defined in the geometry AOCS/Instrument coupling block matrix.

Input image should have a finer resolution than the observation area grid with already applied sampling factor.
For each detector element, the Instrument Projection sub-block projects the pixel array onto the imaging area. The energy received depends on the Instantaneous Field Of View (IFOV) and pixel dimensions. In BIBLOS a constant nominal detector pitch (pixel size) is considered. There is no attenuation considered due to detector pitch. In the following equation, size_pixel is considered equal to size_ref.

\[ L_{\text{pix}} = L_{\text{step}} \cdot \frac{\text{size}_{\text{pix}}}{\text{size}_{\text{ref}}} \]

The method for the resampling is bilinear interpolation.

- Q11(x1,y1), Q12(x1,y2), Q21(x2,y1) and Q22(x2,y2)) are the coordinates of the external inputs.
7.1.1. DETAILED DESCRIPTION

1. "External Image", "scene latitude", "scene longitude", are taken as inputs.
2. There are two possible external image types to be input into the resampling block:
   - External TOC map
   - External Physical parameters map
3. Whole observation area should be covered with external image data. Boundaries of the observation area are transformed from the GEO to the UTM coordinates to check if boundaries of the observation area are included inside. If not the error is returned.
4. Observation area grid points are transformed to the UTM coordinates
5. Values of TOC/physical parameters are calculated for each Observation area grid point by bilinear interpolation of the original TOC/physical parameters map
   - \( Q_{11}(x_1,y_1), Q_{12}(x_1,y_2), Q_{21}(x_2,y_1), Q_{22}(x_2,y_2) \) are the coordinates of the external inputs (TOC/physical parameters)
   - \( f(Q_{11}), f(Q_{12}), f(Q_{21}), f(Q_{22}) \) are the external input values
   - \( P(x,y) \) is the observation point
   - \( f(P) \) is the value of the TOC/physical parameter in the observation area grid

\[
f(x,y) = \frac{1}{(x_2 - x_1)(y_2 - y_1)}(x_2 - x)(y_2 - y) + (x - x_1)(y_2 - y) + (x_2 - x)(y - y_1) + (x - x_1)(y - y_1))
\]
The idea of bilinear interpolation is presented on Figure 7-3:

\[ f(x,y) = \frac{1}{(x_2-x_1)(y_2-y_1)} \left( (x_2-x)(y_2-y) + (x-x_1)(y_2-y) + (x_2-x)(y-y_1) + (x-x_1)(y-y_1) \right) \]

6. The output map is **TOC map in observation area** or **Physical parameters map in observation area** depending on the input map.
7.1.2. FLOW DIAGRAM

Figure 7-4 Resampling block flow diagram

7.1.3. INTERFACES

7.1.3.1. Configuration Parameters

The following configuration parameters are defined.

- Global includes parameters that affect several modules
- Local refers to parameters that affect only this module
- Variable Type is the C data type
### Table 7-1 Configuration Parameters definition for attitude block

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attitude mode</td>
<td>Attitude mode (XLBIBLOS_ATTITUDE_MODE_GEODE蒂C_YAW_STEERING)</td>
<td>string with variable</td>
<td>N.A.</td>
<td>Local</td>
</tr>
</tbody>
</table>

### 7.1.3.2. Inputs

#### Table 7-2 Attitude Block Input Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>External Image</td>
<td>This is the external TOC image or external physical parameters map</td>
<td>File of various-size double array</td>
<td>N.A.</td>
</tr>
<tr>
<td>Scene_lat</td>
<td>Output Acquisition scene matrix of size N x M. If the pass is descending the top row is the northern part of the image, if the pass is ascending the first row is the southernmost point. The first detector element shall be the first defined in the input LOS_instrument file. This matrix contains the latitude geodetic coordinate. N is the number of subpixels along track M is the number of subpixels across track.</td>
<td>File of various-size double array</td>
<td>rad</td>
</tr>
<tr>
<td>Scene_lon</td>
<td>This matrix contains the longitude geodetic coordinate and has the same size and corresponding values than the previous latitude.</td>
<td>File of various-size double array</td>
<td>rad</td>
</tr>
</tbody>
</table>

### 7.1.3.3. Outputs

#### Table 7-3 Attitude Block Output Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOC map in observation area / Physical parameters map in observation area</td>
<td>TOC map in observation area / Physical parameters map in observation area – this file contains the TOC/physical parameters resampled to the observation area</td>
<td>File of various-size double array</td>
<td>N. A.</td>
</tr>
</tbody>
</table>

### 7.1.4. SCOPE AND LIMITATIONS

This block is applicable to all Earth Observation missions, including missions with the following instruments:
- Passive Opticals
- Active Microwaves
- Passive Microwaves
- Active Opticals

### 7.1.5. EXTERNAL LIBRARIES

The libraries used inside this block are:
- Eigen, §11.1
- NetCDF §11.5
- EOCFIs §11.6
- Boost §11.4
- Logger §11.3
7.1.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfiError Error class, [RD.26]. Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception in thrown in case of error and the execution stops. Warnings do not stop the execution.

For the Resampling block the following errors can occur:
- File access: input files not found.
- Errors inside the EOCFI functions.
- Error in memory allocation.

7.2. ATMOSPHERE SIMULATOR BLOCK

The Atmosphere Simulator Block is in charge of the propagation of reflected/emitted light from the surface through the atmosphere for an Earth surface pointing scene.

The execution of an atmospheric RTM for the simulation of the TOA radiance can be computationally very demanding for large size scenes. The option selected for BIBLOS is to generate an atmospheric Look-Up-Table (LUT) with several combinations of atmospheric/illumination/viewing conditions. This is a flexible configuration, as the user can modify or change the LUT with a RTM that suits their mission. The LUT will also be used later in Level-2 Retrieval Module and. The main atmospheric parameters that characterize the optical properties of the atmosphere are Aerosol type, Aerosol Optical Thickness, Water Vapour Column, Carbon Dioxide, and Ozone Content.

libRadtran and MODTRAN

The two most extended Atmospheric RTMs in the remote sensing scientific community are libRadtran, [RD.12], and MODTRAN, [RD.13]. However there are certain differences between these models:
- Primarily libRadtran was created for the purpose of correcting imagery in the UV and VIS ranges of the electromagnetic spectrum. Despite later modifications, its scope is limited to imagery acquired using optical sensors (i.e. UV, VIS, NIR, SWIR and TIR). There is no possibility of correcting imagery acquired using microwave sensors. MODTRAN enables atmospheric corrections for all types of remote sensing imagery (UV to microwave).
- LibRadtran is available via a GNU General Public License. MODTRAN is a proprietary model which is not available for free. It has a proprietary commercial license.
- The atmospheric correction conducted using the MODTRAN model is based on a band approach, in which the atmosphere had been divided into wide layers. This method assumes that the optical properties of the medium are constant within each layer. This is a very fast method, however it can sometimes give a very rough estimate of the atmosphere's properties. A much better and more precise method is a line-by-line approach, which is implemented in the libRadtran model. This model allows the user to select either the band or line-by-line approach.

Based on the above parameters, it can be said that the libRadtran model is a more efficient RTM between these two, when dealing with passive optical sensor data. However one must not limit themselves only to these two models. Modelling of the atmosphere is a very complex process and there is no universal model which would be applicable for all remote sensing data. Most existing RTM's had been designed for specific types of sensors, their spectral resolution, the types of land cover etc.

The first choice for an atmospheric RTM is LibRadTran.
The atmosphere radiative transfer model that will be considered in the simulator is libRadtran ([RD.12]). More accurately, it shall be said that LUTs derived from several runs of libRadtran SW will be used. The libRadtran model allows the input of a great variety of atmosphere, aerosol, clouds, or solar spectra models (and associated variables) and it also allows the retrieval of the radiative transfer output in many different ways.

The following parameters will be covered in the LUTs:

- Sun zenith angle of each area being acquired.
- Observation zenith angle of the acquisition (from the satellite).
- Relative Sun-spacecraft azimuth.
- Surface altitude.
- Surface albedo.
- Aerosol haze, which indicates the aerosol type in the lower 2 km of the atmosphere, will have four possible options: rural type, maritime type, urban type and tropospheric type.
- Aerosol volcan, which indicates the aerosol type above 2 km, could have also four possible options: background aerosols, moderate volcanic aerosols, high volcanic aerosols and extreme volcanic aerosols. Only background and high volcanic cases have been considered.
- Aerosol season. Two seasons are considered: spring-summer and fall-winter.
- Aerosol visibility, which indicates the horizontal visibility in km.
- Precipitable water (in kg/m2), for the scaling of the water vapour profile.
- Advanced users could generate their own LUTs for their use in EIPS (user-defined models for aerosols, atmosphere pressure and temperature, clouds...).

**TOA Radiance calculation**

For flat, lambertian surface the TOA radiance is:

\[
L_{\text{TOA}} = L_{\text{atm}} + \rho'_s E_0 \cos(SZA) T(\cos(SZA)) T(\cos(OZA)) \left(1 - \rho''_s S_{\text{atm}}\right) \pi
\]

\[= \frac{\rho'_s E_0 \cos(SZA) T(\cos(SZA)) T(\cos(OZA))}{(1-\rho''_s S_{\text{atm}})} \]

(1)

Where,

- \(L_{\text{atm}}\) is the intrinsic contribution of the atmosphere to the TOA radiance;
- \(\rho'_s\) is the uniform lambertian TOC reflectance of one pixel (for one spectral band of the reference image);
- \(E_0\) is the extra-terrestrial flux;
- \(\cos(SZA)\) is the cosine of the solar zenith angle;
- \(T(\cos(SZA))\) is the total atmospheric transmission from the sun to the surface, dependant on the cosine of the solar zenith angle;
- \(T(\cos(OZA))\) is the total atmospheric transmission from the surface to the sensor, dependant on the cosine of the observation zenith angle;
- \(S_{\text{atm}}\) is the spherical albedo of the atmosphere.

LibRadtran can be run three times for each combination of observation angles and surface attitude with three different surface reflectances \((\rho'_s = 0, 0.4, 0.8)\). The outputs of these calculations which are saved to LUT are \(L_{\text{TOA}}(\rho_0), L_{\text{TOA}}(\rho_1), L_{\text{TOA}}(\rho_2)\).

Given the values of the \(L_{\text{TOA}}(\rho_0), L_{\text{TOA}}(\rho_1), L_{\text{TOA}}(\rho_2)\) for a specific SZA, OZA, Altitude and wavelength read from the LUT (and interpolated) for the particular pixel, \(L_{\text{atm}}, S_{\text{atm}}, E_0, \frac{cos(SZA) T(\cos(SZA)) T(\cos(OZA))}{\pi}\) values can be calculated as shown below:
• First, the intrinsic contribution of the atmosphere to the TOA radiance is calculated:

\[ L_{\text{atm}} = L_{\text{TOA}}(\rho_0) \]  

(2)

• Than the spherical albedo of the atmosphere

\[ S_{\text{atm}} = \frac{1 - \rho_2}{\rho_1} \left( \frac{L_{\text{TOA}}(\rho_2) - L_{\text{TOA}}(\rho_0)}{L_{\text{TOA}}(\rho_1) - L_{\text{TOA}}(\rho_0)} \right) \]  

(3)

• Next the extraterrestrial flux:

\[ E_0 \cdot \frac{\cos(SZA) \cdot T(\cos(SZA)) \cdot T(\cos(SZA))}{\pi} = \left( \frac{1}{\rho_2} - \frac{1}{\rho_1} \right) \left( \frac{L_{\text{TOA}}(\rho_2) - L_{\text{TOA}}(\rho_0)}{L_{\text{TOA}}(\rho_1) - L_{\text{TOA}}(\rho_0)} \right) \]  

(4)

Where \( \rho_0, \rho_1, \rho_2 \) are the surface reflectances of the 0, 0.4, 0.8 respectively.

• Last step is to include the resulting values of \( E_0 \cdot \frac{\cos(SZA) \cdot T(\cos(SZA)) \cdot T(\cos(SZA))}{\pi}, S_{\text{atm}}, L_{\text{atm}} \) in the \( L_{\text{TOA}} \) equation for given \( \rho'_S \).

Generated LUT’s should include TOA surface reflectances for the TOC reflectances of 0, 0.4, 0.8 with the following parameters discretizations:

• Spectral resolution of
  • Near Infrared to UV: 1 nm in the range of the spectrum between 200 and 1000 nm.
  • Middle Infrared: 10 nm in the range of the spectrum between 1.0 and 2.6 µm.
  • Thermal Infrared: 50 nm in the range of the spectrum between 2.6 and 14 µm.

• Solar zenith angles between 0 and 70 degrees, every 10 degrees.

• Observation polar angles between 0 and 40 degrees, every 10 degrees.

• Relative solar-observation azimuth angles between 0 and 180, every 60 degrees. These values, together with the symmetry between 0-180 degrees and 180-360 degrees allow fitting the data to a polynomial and retrieving the values for another relative azimuth angle.

• Surface altitudes between 0 and 4 km, every 500 m.

LUT generation

Atmospheric radiative transfer model that will be considered in the simulator is LibRadTran. TOA radiance is calculated for the range of variables.

Generated LUT’s should include TOA surface reflectances for the TOC reflectances \( (\rho'_S = 0, 0.4, 0.8) \) with the following parameters discretizations:

• Spectral resolution of
  • Near Infrared to UV: 1 nm in the range of the spectrum between 200 and 1000 nm.
  • Middle Infrared: 10 nm in the range of the spectrum between 1.0 and 2.6 µm.
  • Thermal Infrared: 50 nm in the range of the spectrum between 2.6 and 14 µm.

• Solar zenith angles values of 0°, 10°, 20°, 30°, 40°, 50°, 60°, 70°.

• Observation (viewing) zenith angles between 0° and 40° degrees, every 10°.
Relative solar-observation azimuth angles values 0° to 180°, every 60°. These values, together with the symmetry between 0-180 degrees and 180-360 degrees allow fitting the data to a polynomial and retrieving the values for another relative azimuth angle.

Surface altitudes between 0 and 4 km, every 500 m.

The outputs of these calculations which are saved to LUTs are $L_{\text{TOA}}(\rho_0)$, $L_{\text{TOA}}(\rho_1)$, $L_{\text{TOA}}(\rho_2)$.

Validation

The model described here is validated. The validation of this block shall be in the generation of the LUTs and in the call to the LUTs.

7.2.1. DETAILED DESCRIPTION

Atmosphere Simulator (using LUT)


2. For each pixel the $\rho''$, SZA, OZA, RAA, Altitude are read.

3. For each pixel the multi-linear interpolation is performed from the LUT values to find $L_{\text{TOA}}(\rho_0)$, $L_{\text{TOA}}(\rho_1)$, $L_{\text{TOA}}(\rho_2)$ for a given pixel's parameters:
   - For all parameters that change from pixel to pixel and are tabularized in LUT i.e. SZA, OZA, RAA, Altitude, the first lower and higher values, w.r.t. pixel's nominal parameters values, are found. For all combinations, there are in general $2^4 = 16$ points in the LUTs that surround given pixel's set of parameters. For each of these 16 points sets of $L_{\text{TOA}}(\rho_{i})\lambda$, $L_{\text{TOA}}(\rho_{1})\lambda$, $L_{\text{TOA}}(\rho_{2})\lambda$ values for each wavelength are read.
   - The imager band is taken into account. For each of the 16 points values of $L_{\text{TOA}}(\rho_{i})\lambda$, $L_{\text{TOA}}(\rho_{1})\lambda$, $L_{\text{TOA}}(\rho_{2})\lambda$ are summed along the range of wavelengths composing $[L_{\text{TOA}}(\rho_{i})\lambda; L_{\text{TOA}}(\rho_{1})\lambda; L_{\text{TOA}}(\rho_{2})\lambda](\lambda_{\text{min}}:\lambda_{\text{max}})$ values for the whole band.
     $L_{\text{TOA}}(\rho_{i})\big|_{(\lambda_{\text{min}}:\lambda_{\text{max}})} = \sum_{\lambda=\lambda_{\text{min}}}^{\lambda_{\text{max}}} L_{\text{TOA}}(\rho_{i})\lambda$
   - 4-dimensional linear interpolation is performed over the 16 points to find three $L_{\text{TOA}}(\rho_{i})\big|_{(\lambda_{\text{min}}:\lambda_{\text{max}})}$ values for a given pixel's parameters.

4. The $E_0 \times \frac{\cos(\text{SZA})}{\pi} \times T(\cos(\text{SZA})) \times T(\cos(\text{SZA}))$, $S_{\text{atm}}$, $L_{\text{atm}}$ are calculated using equations (2), (3) and (4).

5. $L_{\text{TOA}}$ for the given pixel’s TOC reflectance $\rho''$ is calculated using equation (1).

6. Results for all pixels are saved in one file. The output is TOA map w/o clouds.
7.2.2. FLOW DIAGRAM

Figure 7-5 Atmosphere simulator block flow diagram

7.2.3. INTERFACES

7.2.3.1. Configuration Parameters

The following configuration parameters are defined.

- Global includes parameters that affect several modules
• Local refers to parameters that affect only this module
• Variable Type is the C data type

Table 7-4 Configuration Parameters definition for attitude block

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imager_band</td>
<td>This variable describes the bandwidth of the current imager. Two values are: Central wavelength of the band and bandwidth.</td>
<td>2x1 double array</td>
<td>m</td>
<td>Global</td>
</tr>
</tbody>
</table>

7.2.3.2. Inputs

Table 7-5 Attitude Block Input Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOC map in observation area</td>
<td>This is the external TOC image or external physical parameters map</td>
<td>File of various-size double array</td>
<td>N.A.</td>
</tr>
<tr>
<td>Scene_lat</td>
<td>Output Acquisition scene matrix of size N!!M. If the pass is descending the top row is the northern part of the image, if the pass is ascending the first row is the southernmost point. The first detector element shall be the first defined in the input LOS_instrument file. This matrix contains the latitude, longitude, and altitude geodetic coordinate. N is the number of subpixels along track M is the number of subpixels across track</td>
<td>NetCDF file, double values saved in double precision</td>
<td>rad</td>
</tr>
<tr>
<td>Scene_SZA</td>
<td>This matrix contains the Sun Zenith Angle. It has the same size and corresponding values than the previous latitude.</td>
<td>NetCDF file, double values saved in double precision</td>
<td>rad</td>
</tr>
<tr>
<td>Scene_OZA</td>
<td>This matrix contains the Observation Zenith Angle. It has the same size and corresponding values than the previous latitude.</td>
<td>NetCDF file, double values saved in double precision</td>
<td>rad</td>
</tr>
<tr>
<td>Scene_RAA</td>
<td>This matrix contains the Relative Azimuth Angle. It has the same size and corresponding values than the previous latitude.</td>
<td>NetCDF file, double values saved in double precision</td>
<td>rad</td>
</tr>
<tr>
<td>TOA_LUT</td>
<td>Look-up-table containing: $L_{TOA}(\rho_1)$, $L_{TOA}(\rho_2)$, $L_{TOA}(\rho_3)$ for the discretized range of parameters.</td>
<td>Group of XML-files</td>
<td>N.A.</td>
</tr>
</tbody>
</table>

7.2.3.3. Outputs

Table 7-6 Attitude Block Output Parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOA map in observation area / Physical parameters map in observation area</td>
<td>TOC map in observation area / Physical parameters map in observation area – this file contains the TOC/physical parameters resampled to the observation area</td>
<td>File of various-size double array in NetCDF</td>
<td>N.A.</td>
</tr>
</tbody>
</table>

7.2.4. SCOPE AND LIMITATIONS

This block is applicable to Earth Observation missions, including missions with the following instruments:
• Passive Opticals
• Active Opticals

The block is applicable only to missions within the constraints below:

• Wavelength spectrum between 200 and 14000 nm
• Solar zenith angles between 0 and 70 degrees
• Observation polar angles between 0 and 40 degrees
• Surface altitudes between 0 and 4 km

7.2.5. EXTERNAL LIBRARIES

The libraries used inside this block are:

• Eigen, §11.1
• NetCDF §11.5
• EOCFIs §11.6
• Boost §11.4
• Logger §11.3

7.2.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfiError Error class, [RD.26].

Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception in thrown in case of error and the execution stops. Warnings do not stop the execution.

For the Atmospheric block the following errors can occur:

• File access: input files not found.
• Errors inside the EOCFI functions.
8. INSTRUMENT MODULE

The instrument module simulates the conversion of the energy arriving at the instrument (radiance), to digital numbers saved to file. It is composed of two blocks for a basic simulation chain for a passive optical instrument, the spatial, and the radiometric blocks:

Spectral, Spatial and Radiometric resolution

The architecture of the simulator allows to input as many bands with as fine spectral resolution as wanted, the radiometric resolution (bit-depth), the integration time of the instrument and the detector pitch (pixel size). This allows to model the following types of passive optical imagers:

- **Multi-band imagers**: Typically these have bands with 50-70nm of bandwidth in the Red, Blue, Green and Near-Infrared section of the electromagnetic spectrum, or Panchromatic. The spatial resolution is several meters for low-Earth orbits.
- **Radiometers**: Similar to Multi-band imagers but with a higher radiometric resolution, i.e. the number of shades is higher, defined by the bit depth. This comes at the expense of a lower spatial resolution and/or spectral resolution.
- **Instruments with finer spectral resolution**: The current version of BIBLOS allows to define as many detector lines and with as small bandwidth as desired.

Spectrometers are not covered in the current version of BIBLOS. Spectrometers have a high spectral resolution (of around 1nm). The architecture for spectrometer simulation is different to the multi-band imagers and radiometers, as a ‘cub’ of information is acquired each sampling time.
8.1. SPATIAL BLOCK

The Spatial Block simulates the recorded pixel information when the TOA radiance arrives to the instrument detector elements. This Block is applicable to Imagers and Radiometers.

The Spatial Block performs the spatial resampling to the final ground sampling distance, taking into account the Satellite instrument Modulation Transfer Function and spatial non-uniformity sources.

The process is characterized by the convolution of the spectral surface information with filter functions along and across orbit representing the sensor specific Point Spread Functions (PSF). This incorporates optical, detector, vibration and spacecraft motion characteristics.

The Spatial Block simulates the spatial domain of the image, which is correlated with the instrument type (whiskbroom, pushbroom or frame). Pushbroom and frame instruments can be considered as particular types of whiskbroom instruments in which the acquisition of a group of pixels (line in pushbroom instruments or 2D array in frame instruments) is taken at the same time.

In the optical phase, there are several errors to be taken into account. Some of them are:

- High frequency vibrations or Jitter.
- Pixelation
- Smearing
- Time Delay Integration
- Straylight

The steps to process an image block at the optical stage are:

2. Perform a 2D Fourier Transform
3. Multiply the image in Fourier domain by the two-dimensional MTF
4. Perform the inverse Fourier Transform

8.1.1. DETAILED DESCRIPTION

Memory Management

The first point in the architectural design is the memory management. As is described in §5.4.4 typical images cannot be held in its entirety in memory for an average workstation. Therefore the scene to be processed is divided into smaller units called Squares. Each square is processed independently and stored to file to complete the whole scene. The process is described in detail here. The process in the spatial block of each Square is described in the Spatial Aberrations that are simulated, and the Spatial Resampling.

The Scene is divided into Squares. The Scene is processed in a double loop across and along track for every square as calculated by §5.4.8.
The input to the Instrument Module is the scene resampled at subpixel level according to the scale factor (§5.4.5). Additionally, there is an extra margin of area processed that is later discarded during the spatial resampling. The assignation of the subpixel information to each Square is represented in the following image:

Figure 8-2: Memory Management - Division of the Scene in Squares

Figure 8-3: Double loop in the ACT and ALT directions to process each square.
Analytically, for a Square in along track (ALT) position \( ii \) and across track (ACT) position \( jj \) the four corners of the square are:

- \( \text{iPixAct} = \text{iact} \times (\text{input_square_size} - 2 \times \text{square_margin}) \)
- \( \text{iPixAlt} = \text{ialt} \times (\text{input_square_size} - 2 \times \text{square_margin}) \)

Where
- \( \text{iPixAct} \) is the index of first subpixel in the ACT direction
- \( \text{iPixAlt} \) is the index of first subpixel in the ALT direction
- \( \text{ialt} \) is the number of the Square ALT, and it goes from 0 to the maximum number of squares ALT
- \( \text{iact} \) is the number of the Square ACT, and it goes from 0 to the maximum number of squares ACT
- \( \text{square_size} \) is the size of the square in subpixels, before the resampling
- \( \text{square_margin} \) is the extra border that is processed and later discarded in the resampling

Note that in C, C++ index starts on 0. If this is implemented in Matlab for example the first index is 1, so \( jj\_start \) and \( ii\_start \) have +1 added.

With this index the image is loaded from file into memory.

**Merging of the bands of the TOA image to the instrument bands.**
The input image typically has a higher spectral resolution than the instrument. So if the imager has 4 or 5 bands (typically multispectral read, blue, green, near infrared, panchromatic) the input image may have 15-20 or even more bands that cover the same portion of the electromagnetic spectrum. In the Scene Generator each band is propagated through the atmosphere to obtain the TOA image for each band of the input image.

The first step of the spatial block is to merge the bands to the instrument acquisition band. That is, if for example the detector line acquires in the 500-600nm range (colour green), the energy of the input bands that fall in that segment is added.

![Figure 8-5: Characterization of spectral bands of a reference image (dotted black lines) with respect to some of the typical imager bands (multispectral blue, green, red, panchromatic)](image)

Taking into account the former remarks, the merging of radiometric information from reference image narrow bands to the bands that correspond to the actual instrument band is based on the following equation:

\[
L_{SEOS\_band} = \sum L_{\lambda\_i} \cdot \Delta\lambda_{uni\_i} \cdot R(\Delta\lambda_{uni\_i}) + \sum \frac{L_{\lambda\_j} + L_{\lambda\_l}}{2} \cdot \Delta\lambda_{over\_j} \cdot R(\Delta\lambda_{over\_j}) + \sum \frac{L_{\lambda\_k} + L_{\lambda\_l}}{2} \cdot \Delta\lambda_{gap\_kl} \cdot R(\Delta\lambda_{gap\_kl})
\]

Where:

- \(L_{SEOS\_band}\) is the TOA radiance corresponding to a certain instrument band.
- \(L_{\lambda\_i}\) is the spectral TOA radiance for a certain reference image band.
- \(\Delta\lambda_{uni\_i}\) is the spectral width of reference image band \(i\) that falls within the considered instrument band and has no overlap with other bands of the reference image.
- \(\Delta\lambda_{over\_j}\) is the spectral width of reference image band \(i\) that overlaps with the spectral width of the reference image band \(j\).
- \(\Delta\lambda_{gap\_kl}\) is the spectral width of the instrument band that is not covered by any of the bands of the reference image, and is limited by reference image bands \(k\) and \(l\).
\( R(\Delta \lambda) \) is the peak-normalised spectral response of the instrument band, averaged over a certain spectral width, and is an input to the simulator.

**Radiance to Irradiance Conversion**

Radiance \([W \, m^{-2} \, Sr^{-1}]\) emitted by the ground has to be converted to irradiances \([W \, m^{-2}]\) on the focal plane. The difference is that the first is the power emitted or received per solid angle while the second is emitted or received power computed for a certain solid angle.

The calculation of the irradiance from the radiance is done following this equation:

\[
I = T_R \cdot L \cdot \frac{\pi}{4 \cdot f^2} \cdot (D_{pupil}^2 - D_{obstruction}^2)
\]

Where:
- \(I\) is the irradiance
- \(L\) is the Radiance
- \(D_{pupil}\) is the pupil diameter
- \(D_{obstruction}\) is the diameter of the central obstruction
- \(h\) is the altitude of the satellite
- \(f\) is the focal length of the telescope
- \(T_R\) is the Transmittance (for non-transparent objects \(T_R < 1\))

**Spatial Aberrations – Application of an external MTF**

The response in the spatial domain of the camera is modelled via the application of an MTF.

1. Calculation of the Spatial frequencies, in the along and across track directions.
   The spatial frequencies are the distance in subpixels.
   The Nyquist frequency for the detector is based on the detector pitch (pixel size):
   \[
f_{\text{nyq, pix}} = \frac{1}{(2 \cdot \text{pixel_size})}
\]
   Where
   - \(f_{\text{nyq, pix}}\) is the Nyquist frequency for the detector element (pixel)
   - \(\text{pixel_size}\) is the size of the pixel of the detector line

   For an accurate effect simulation, the pixel is artificially simulated at subpixel level. The scale factor determines how many subpixels there are in each pixel. The Nyquist frequency at subpixel level is:
   \[
f_{\text{nyq, subpix}} = \frac{1}{(2 \cdot \text{subpixel_size})}
\]
   Where
   - \(f_{\text{nyq, subpix}}\) is the Nyquist frequency for the subpixel
   - \(\text{subpixel_size}\) is the size of the pixel of the detector line divided by the scale factor

   The spatial frequencies is the distance in m-1 for each subpixel sampling:
   The image is processed in squares of size \(\text{input_square_size} \times \text{input_square_size}\) (see §5.4.4) so it is enough to calculate the spatial frequencies and the MTF in 2D for a distance of \(\text{sqrt}(2)\) of the \(\text{input\_square\_size}\) (the diagonal distance).

   ```
   while (!end_loop) {
     // Increases a step amount
     positive_freq_vector.push_back(positive_freq_vector[ifr - 1] + step);
   }
   ```
if ((positive_freq_vector[ifr]) > sqrt(2.0)) {
    end_loop = true;
}

// Increase index
ifr++;
}

for (unsigned int ifr = 1; ifr < positive_freq_vector.size(); ifr++) {
    positive_freq_vector[ifr] = (positive_freq_vector[ifr] * f_nyq_subpix);
}

2. Calculation of the MTF in 1D in the across and along track directions at the spatial frequencies of the detector line.

The user inputs the MTF in 1D in the across track and along track directions (see [PD.7] for example and format). The MTF in 1D is defined at given spatial frequencies, so the first step is to interpolate the MTF to the spatial frequencies of the detector line at subpixel level.

3. Calculation of the MTF in 2D based on reading the MTF in 1D (defined by the user), in the across and along track directions.

The MTF in 2D is calculated as the product of the MTF in 1D in the across and along track directions, with the pixel distance from the centre (the centre value of the MTF is 1):

// Pixel coordinates where the zero frequency is located (centre of the matrix)
int alt_center = (size_alt / 2);
int act_center = (size_act / 2);

// Fill 2D MTF matrix
for (unsigned int ialt = 0; ialt < size_alt; ialt++) {
    for (unsigned int iact = 0; iact < size_act; iact++) {

// Calculating distance in pixels (or vector index) to the zero frequency
// located at alt_center, act_center
int dalt = (int)abs((float)ialt - (float)alt_center);
int dact = (int)abs((float)iact - (float)act_center);

mtf2d->values(ialt, iact) = mtf1d_alt[dalt] * mtf1d_act[dact];

Figure 8-7: MTF in 2D calculated from the MTFs in 1D for the across and along track dimensions. The MTF in 2D is only calculated for 1024x1024 samples of subpixels because that is the size of image processed in each step. The MTF in 1D for the ACT and ALT directions in this example are the same, so in this case the MTF2D is symmetrical in the X and Y axis.

4. Perform Fast Fourier transform (FFT)
The FFT is done to obtain the image in the frequency domain. This is done by calling the KISS FFT (see § 11.2) library. pixels_data is the image in the spatial domain and pixels_data_fft is the image in the spectral domain, the FFT is done following these steps:

Stores the image pixels_data in the necessary format, i.e. 1 dimensional array of type fftw_complex with size input_square_size x input_square_size

fftw_complex *in = NULL;
fftw_plan p;

Allocate one-dimensional contiguous array of dimension input_square_size^2
in = (fftw_complex*) fftw_malloc(sizeof (fftw_complex) * input_square_size^2);

Create a plan, an object containing the necessary parameters
p = fftw_plan_dft_2d(input_square_size, input_square_size, in, pixel_data_fft, FFTW_FORWARD, FFTW_ESTIMATE);

Turn the 2D array "pixels_data" into C-order (row major) 1D array of
complex numbers (the imaginary part is filled with zeros)
The input image is real so the [1] is completed with 0s
for (int i = 0; i < input_square_size; i++) {
    for (int j = 0; j < input_square_size; j++) {
        in[j + input_square_size * i][0] = pixels_data[i][j] * pow(-1, (i+j));
        in[j + input_square_size * i][1] = 0;
    }
}

Perform FFT. It takes the input from in and saves the result in pixel_data_fft.
fftw_execute(p);

De-allocate memory
fftw_destroy_plan(p);
fftw_free(in);

Finally, save the image in the spectral domain in case the user wishes to apply other MTFs.

5. Once the image is in the spectral domain, multiply the MTF times the image in the spectral domain (Fourier transform). The Fourier transform of the image has a real and imaginary parts and the MTF has only a real part.

for (i = 0; i < input_square_size; i++) {
    for (j = 0; j < input_square_size; j++) {
        // Real part
        pixel_data_fft[j + input_square_size * i][0] =
        MTFValue[i][j] * pixel_data_fft[j + input_square_size * i][0];

        // Imaginary part
        pixel_data_fft[j + input_square_size * i][1] =
        MTFValue[i][j] * pixel_data_fft[j + input_square_size * i][1];
    }
}

The image is saved in the spectral domain in case the user wishes to apply other MTFs. Other effects modelled with MTFs can be added at this point with the same function (Jitter, OTF, pixelation, smearing, etc).

6. After the MTF is applied the image is converted to the spatial domain with an Inverse Fourier Transform. This is also done with calls to the fftw library.

Allocate output array
fftw_complex *out = (fftw_complex*) fftw_malloc(sizeof (fftw_complex) * input_square_size^2);

Create plan in mode FFTW_BACKWARD. Object containing necessary parameters
fftw_plan p;
p = fftw_plan_dft_2d(input_square_size, input_square_size, pixel_data_fft, out, FFTW_BACKWARD, FFTW_ESTIMATE);

Perform inverse FFT
fftw_execute(p);

normalization_factor = input_square_size^2;
for (i = 0; i < input_square_size; i++) {
    for (j = 0; j < input_square_size; j++) {

pixels_data[i][j] = out[j+input_square_size*i][0];

After doing a FFT or IFFT transform the first and third quadrants and also the second and fourth quadrants are swapped. Due to the DFT properties it is possible to reorder the quadrants indirectly by multiplying all values by \((-1)^{(i+j)}\). (See for example http://www.fftw.org/faq/section3.html#centerorigin). Also the two dimensional matrix is stored in a one-dimensional array consistently with the FFTW library representation of multidimensional arrays.

\[
\text{out}[j+\text{input_square_size}*i][0] = \text{out}[j+\text{input_square_size}*i][0] \times \text{pow}(-1,(i+j));
\]

\[
\text{out}[j+\text{input_square_size}*i][1] = \text{out}[j+\text{input_square_size}*i][1] \times \text{pow}(-1,(i+j));
\]

Do absolute value and save in pixel_data. Due to numerical reasons complex numbers with small imaginary component might appear for some pixels. The phase is meaningless and therefore the absolute value is taken

\[
\text{pixels_data}[i][j] = \sqrt{(\text{out}[j+\text{input_square_size}*i][0])^2 + (\text{out}[j+\text{input_square_size}*i][1])^2};
\]

Normalization needed according to the normalization criteria of the FFTW library.

\[
\text{pixels_data}[i][j] = \text{pixels_data}[i][j] / \text{normalization_factor};
\]

De-allocate memory

\[
\text{fftw_destroy_plan}(p);
\]

\[
\text{fftw_free(out)};
\]

Spatial Resampling

Spatial resampling is the process of averaging the information of several subpixels to pixel level. The Square margin is removed at this stage. After the image is at pixel level.

Input Square: 1024 x 1024 subpixels

Output Square: 300 x 300 pixels

Figure 8-8: Size of the Square before and after the spatial resampling.

1. Reservation of memory for an auxiliary array aux_image the size of the output square and initialization to zeros.
2. Check that the relation between input square size, the output square size, the square margin and the scale factor is met. See §5.4.7.
3. Loop on all indices of the large image. For each index The corresponding index in the output image where the value shall go is calculated. The margin is removed

```c
for (int ii_input = 0; ii_input < input_img_size; ii_input++) {
    // Calculate i index in the reduced image
    get_index(ii_input, ii_output, i_inside_square);
    for (int jj_input = 0; jj_input < input_img_size; jj_input++) {
        // Calculate j index in the reduced image
        get_index(jj_input, jj_output, j_inside_square);
        // If the subpixel is located inside the square margin it is lost.
        if (i_inside_square == true && j_inside_square == true) {
            aux_image[ii_output][jj_output] = aux_image[ii_output][jj_output] + input_image[ii_input][jj_input];
        }
    }
}
```

Where
- `input_image` is the scene at subpixel level
- `aux_image` is the auxiliary image that is the size of the output image
- `ii_input` and `jj_input` are the indices of access to the input image at subpixel level
- `input_img_size` is the size of the input image
- `ii_output` and `jj_output` are the indices of access to the output image at pixel level
- `i_inside_square` and `j_inside_square` are Booleans that say whether a subpixel is part of the square margin or not

**get_index function**

This function calculates the corresponding index in the output image (pixel level) for the input image (subpixel level). It also calculates whether a subpixels is in the square margin or not.

```c
index_output = floor((index_input - square_margin) / scale_factor);
```

Calculation to see whether the subpixel is inside the square margin or not. If it is inside the margin (`inside_square == false`) it is cut.

```c
if (index_output >= 0 && index_output < output_square_size) {
    inside_square = true;
} else {
    inside_square = false;
    index_output = 0;
}
```

4. In `aux_image` the information of all the subpixels that correspond to a particular pixel is stored. This is now averaged by dividing it by the square of the scale factor.

```c
for (ii_output = 0; i < output_img_size; i++) {
    for (jj_output = 0; j < output_img_size; j++) {
        output_image[i][j] = aux_image[i][j] / (scale_factor^2);
    }
}
```

5. Release memory for the auxiliary array.

**Write to file**

The output image is in binary format. To write to file it is necessary to write a whole line of squares across track at the same time.
1. When a new line of squares along track is processed, memory is reserved for an auxiliary array of the size of the whole line of squares, RowOfSquares. It is initialized to zeros. RowOfSquares is the size of output_img_size times (output_img_size* nblocksACT). Where nblocksACT is the number of input across track in the Scene.

2. Each square is processed.

3. The square is then saved to memory in the line of blocks.

   \[
   \text{ACT\_square\_position} = (\text{nSquaresACT} - \text{jj\_square}) \times \text{output\_img\_size};
   \]

   for \( ii = 0; ii < \text{output\_img\_size}; ii++ \) {
     for \( jj = 0; jj < \text{output\_img\_size}; jj++ \) {
       RowOfSquares[\( ii \)] [\( jj + \text{ACT\_square\_position} \) ] = output_image[\( ii \)] [\( jj \)];
     }
   }

   Where
   - ACT\_square\_position is the offset in across track to save the input.
   - nSquaresACT is the number of input across track in the Scene
   - jj\_square is the index of the Square being processed
   - RowOfSquares
   - ii\_output and jj\_output are the indices of access to the output image at pixel level
   - i\_inside\_square and j\_inside\_square are Booleans that say whether a subpixel is part of the square margin or not

4. When the line of blocks is finished (jj\_square == nSquaresACT) it is written to file.

5. The memory is released.
8.1.2. FLOW DIAGRAM

![Flow Diagram]

Figure 8-10 Spatial block flow diagram

8.1.3. INTERFACES

8.1.3.1. Configuration Parameters

The following configuration parameters are defined:

- Global includes parameters that affect several modules
- Local refers to parameters that affect only this module
- Variable Type is the C data type, from [RD.10]

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number_of_pixels</td>
<td>Number of pixels in the detector line</td>
<td>Unsigned int</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Scale factor</td>
<td>The pixel shall be divided into a number of pixels equal to the square of the scale factor. If the resampling factor is 3 for example, there are 9 subpixels per pixel imaged.</td>
<td>Unsigned integer</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Square margin</td>
<td>Number of Subpixels that are simulated in the processing of each square in the spatial block. Each area of image processed shall have an excess margin of image that will be cut in the spatial block. This area is needed for adjacency calculations and for border effects. The square margin is related to the scale factor, input square size (to the spatial block) and output square size (to the spatial block), with this relation: ((\text{Input square size} - 2 \times \text{square margin})/(\text{scale factor}) = \text{output square size}). For example, if scale factor = 3; square margin = 62; input square size = 1024 and input square size = 300. The following relationship is met: ((1024-2*62)/3=300).</td>
<td>Unsigned integer</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
<tr>
<td>Input square size</td>
<td>INPUT Size of the square that are processed in the spatial block. It is only used in this block to verify that the relation between the input square size, output square size, square margin and</td>
<td>Unsigned integer</td>
<td>Dimensionless</td>
<td>Global</td>
</tr>
</tbody>
</table>
### 8.1.3.2. Inputs

**Table 8-2 Inputs definition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOA Scene in radiance</td>
<td>Top of Atmosphere scene in radiances, output of the Scene Generation Module</td>
<td>Double array [PxQ] in NetCDF</td>
<td>[W m(^{-2}) Sr(^{-1})]</td>
</tr>
<tr>
<td>MTF</td>
<td>Path to the XML MTF file</td>
<td>Char</td>
<td>Dimensionless</td>
</tr>
</tbody>
</table>

### 8.1.3.3. Outputs

**Table 8-3 Outputs definition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type/Format</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene in Irradiance</td>
<td>Scene in irradiance, with spatial resampling and spatial aberrations.</td>
<td>Double array [MxN] in NetCDF</td>
<td>[W m-2]</td>
</tr>
</tbody>
</table>

### 8.1.4. SCOPE AND LIMITATIONS

This block is applicable to Earth Observation missions, including missions with the following instruments:

- Passive Opticals

This Block is applicable to Imagers and Radiometers.

### 8.1.5. EXTERNAL LIBRARIES

The libraries used inside this block are:

- Eigen, §11.1
- NetCDF §11.5
- EOCFIs §11.6
- Boost §11.4
- Logger §11.3
8.1.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfiError Error class, [RD.26]. Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception is thrown in case of error and the execution stops. Warnings do not stop the execution.

For the Spatial block the following errors can occur:

- File access: input files not found.
- Errors inside the EOCFI functions.
- Consistency check function: if the square size before and after resampling, margin and scale factor are not coherent, the simulation stops printing an error message printing the values and explaining the cause. See §5.4.7 for more information.

8.2. RADIOMETRIC BLOCK

The Radiometric Block models the transformation of the energy to the output digital number. It also models the errors that affect the radiometry of the instrument. The radiometric block processing is divided into several steps:

- **Irradiance to photons conversion**
  The image in irradiance has to be converted to number of photons taking into account the photon energy of the band. The central wavelength of the band is used to convert irradiance to photons.

- **Photonic Noise**
  Photonic noise is a noise associated with the particle nature of light. For a level of light irradiance stimulation, with equal interval of exposure time, the number of measured photons will not be the same. This noise models that fundamentally random effect.

  According to the laws of quantum physics the number of photons follows a Poisson distribution. For large numbers the Poisson distribution follows the Gaussian distribution. Given that the typical values of photons are high it will be modelled with a Gaussian distribution.

- **PRNU (Photo Response Non-Uniformity)**
  The PRNU models the imperfect conversion from energy to electrons for each detector elements. Two identical detector elements with the same stimulation will not give the same readout. This effect is modelled with the PRNU. It is applied as a multiplicative factor calculated for each pixel to the photons to electrons conversion with the Quantum Efficiency.

- **Photons to electrons conversion**
  Ideally each photon reaching the detector element is converted in one electron. In practice however not all photons are converted to electrons. The Quantum Efficiency (QE) defines the conversion rate. This parameters varies with each spectral channel.

- **Dark Signal, Dark Signal Non Uniformity**
  The Dark Signal is the output signal in the absence of any illumination for a given temperature and integration time. If there is no external illumination, in the ideal case the readout should be zero, a pure black image. The Dark Signal is present with and without illumination. The Dark signal is also referred to as thermal noise, as it is the consequence of the generation of electrons in the semiconductor due to temperature. Temperature changes are noticeable for orbit periods, in the order of hours. Typical images in the simulators are of seconds, so the working temperature is taken as
constant. The Dark Signal reference value for the working temperature is read as an input in electrons.

The Dark Signal Non Uniformity (DSNU) models the per pixel variation of the Dark Signal residual noise. The DSNU is an additive factor modelled with a normal distribution configurable by the user.

- **Dark noise, Thermal noise**
  The Dark noise is statistical variation in the number of electrons thermally generated within the pixel. It is temperature dependent and lower temperatures indicate less noise.

- **Electrons to Digital Numbers Conversion**
  At this stage, the image is in number of electrons with all detection effects already added. Though the electronics is complex, it can be modelled in three steps:
  - **Electrons to voltage conversion**: It is the process of clocking out the electrons from the CCD detector elements, charge transmission and conversion into an analogue voltage.
  - **Voltage Amplification**: Amplification of the signal.
  - **Signal Digitalisation**: Digitalisation to obtain digital counts.

- **Defective Pixels**
  Malfunction of detector elements in the CCD is modelled at this stage. There are two types of defective pixels:
  - "Dead" pixels permanently showing a low value close to zero
  - "Hot" pixels permanently showing a high value

### 8.2.1. DETAILED DESCRIPTION

**Memory Management, reading and writing to file.**

The memory management is the radiometric blocks is similar to the spatial block but less complex. The Scene is processed in squares and there is a double loop in the across track and along track direction as shown in Figure 8-2 and Figure 8-3. To access the image as produced by the spatial block, the four corners of each square is calculated (with respect to the spatial block, it is simpler given that at this stage the spatial resampling has been done and there is no square margin).

Analytically, for a Square in along track (ALT) position $ii$ and across track (ACT) position $jj$ the four corners of the square are:

\[
\begin{align*}
jj\_start &= \text{output\_square\_size} \times (jj - 1) \\
jj\_end &= jj\_start + \text{output\_square\_size} -1 \\
ii\_start &= \text{output\_square\_size} \times (ii - 1) \\
ii\_end &= ii\_start + \text{output\_square\_size} -1
\end{align*}
\]

Where

- $jj\_start$ is the index of first subpixel in the ACT direction
- $jj\_end$ is the index of the last subpixel in the ACT direction
- $ii\_start$ is the index of first subpixel in the ALT direction
- $ii\_end$ is the index of the last subpixel in the ALT direction
- $ii$ is the number of the Square ALT, and it goes from 1 to the maximum number of squares ALT
- $jj$ is the number of the Square ACT, and it goes from 1 to the maximum number of squares ACT
- $\text{square\_size}$ is the size of the square in subpixels, before the resampling
- $\text{square\_margin}$ is the extra border that is processed and later discarded in the resampling
Note that in C, C++ index starts on 0. If this is implemented in Matlab for example the first index is 1, so \( jj_{\text{start}} \) and \( ii_{\text{start}} \) have +1 added.

With this index the image is loaded from file into memory.

Writing to file is done in the same way than described in the Spatial Block. After one line ACT of square is processed, it is written to file, see Figure 8–9.

**Irradiance to Photons Conversion**

1. "central wavelength", "fill_factor_x", "fill_factor_y", "pixel_size", “Full Well Capacity band” and “image in irradiance” are taken as inputs.
2. Photon Energy \( E_{\text{photon,k}} \) is calculated:
   \[
   E_{\text{photon,k}} = \frac{hc}{\lambda_k}
   \]
   where:
   - \( h \) – Planck’s constant
   - \( c \) – speed of light in vacuum
   - \( \lambda_k \) – central wavelength of the band
3. Energy Conversion Factor is calculated:
   \[
   ECF = \frac{\text{fill_factor}_X \times \text{fill_factor}_Y \times \text{pixel_size}^2 \times \text{integration_time}}{E_{\text{photon,k}}}
   \]
   where:
   - \( \text{fill_factor}_X \) – filled fraction of detector element in X direction
   - \( \text{fill_factor}_Y \) – filled fraction of detector element in Y direction
   - \( \text{pixel_size} \) – length of the detector element [m]
   - \( \text{integration_time} \) – sampling time of the instrument (integration time) [s]
4. Finally the number of photons received by each \( j,i \) pixel for \( k \)-th band is
   \[
   I_{ij}^{hv} = I_{ijk} \times ECF
   \]
   and as the number of photons must be a natural number:
   \[
   I_{ij}^{hv'} = \text{round} \left( I_{ij}^{hv} \right)
   \]
   Where:
   - \( I_{ij} \) – is image in irradiance
   - \( ECF \) – is the energy conversion factor
5. As the number of photons obtained from the detector element cannot be higher than the Full Well Capacity (FWC) for a given detector, values higher than the FWC are cropped to that:
   \[
   I_{ij}^{hv} = \begin{cases} 
   I_{ij}^{hv'} & \text{for} \quad I_{ij}^{hv} < \text{FWC}_k \\
   \text{FWC}_k & \text{for} \quad I_{ij}^{hv} \geq \text{FWC}_k 
   \end{cases}
   \]
   Where:
   - \( \text{FWC}_k \) – is the full well capacity for \( k \)-th band
Photonic Noise

This noise follows a Poisson distribution. For large numbers the Poisson distribution follows the Gaussian distribution. Given that the typical values of photons are high it will be modelled with a Gaussian distribution. Very low numbers of photons shall be skipped (via configuration parameter Min_photon_for_photonic_noise, default value is 10 photons or under

1. Calculation of the Photonic Noise
   If L is the number of photons, the photonic noise is calculated as a Gaussian distribution for each point with mean of L and variance of the square root of L (this is consequence of the property of the Poisson distribution, where the mean and the variance are the same).

2. Addition of the Photonic Noise
   The photonic noise is added to each pixel in the image.

Photo Response Non Uniformity (PRNU)

The PRNU models the effect of how different CCD elements respond differently to the stimulation of energy. It is the ratio between the optical power input on the pixel versus the electrical power output. Two pixels illuminated with the same input power shall produce different electrical outputs.

It is modelled as a multiplicative factor that affects the ideal conversion from power to electrons. The multiplicative factor is modelled with a normal distribution and user configurable mean and standard deviation for the whole CCD. It is calculated for each pixel.

1. The mean and standard deviations to model the gain of the PRNU are taken as an input.

2. Calculation of the PRNU as:

   \[ PRNU_{ijk} = \mu + \sigma \cdot \text{randn} \]

3. The PRNU gain is stored and multiplied to the photons to electrons conversion.

Photons to electrons conversion

1. “Quantum efficiency” is taken as an input.

2. Number of electrons (number of photons converted to electrons) is calculated as:

   \[ I_{ijk}^e = \text{round} \left( I_{ijk}^h \cdot q_{eff}^k \right) \]

where:

   \( q_{eff}^k \) – is a quantum efficiency for k-th band, which describes the conversion rate between photons and electrons for the detector

3. If the PRNU has been selected, it is added at this stage as a multiplicative factor.

Dark Signal Non Uniformity (DSNU)

The DSNU is an additive factor modelled with a normal distribution configurable by the user with mean and variance for the whole CCD. It is calculated for each pixel.

1. The mean and standard deviations to model the gain of the DSNU is taken as an input. The mean of the DSNU is the reference Dark Signal for the working temperature.
2. Calculation of the DSNU per pixel as:

\[ DSNU_{ijk} = \mu + \sigma \cdot \text{randn} \]

3. The Dark Signal is added to the image in electrons.

**Electronic Noise**

The Electronic Noise is modelled with a normal distribution, the variance of which is calculated according to the following formula:

\[ \sigma_{\text{electronic\_noise}} = \sqrt{\sigma^2_{\text{readout}} + \sigma^2_{\text{dark\_noise}}} \]

Where:

- \( \sigma_{\text{electronic\_noise}} \) – Electronic Noise variance
- \( \sigma_{\text{readout}} \) – Readout Noise variance
- \( \sigma_{\text{dark\_noise}} \) – Dark Noise variance

The Readout Noise variance is configurable by the user and the Dark noise variance is calculated according to the following equation:

\[ \sigma_{\text{electronic\_noise}} = \sqrt{\text{dark\_signal} + \text{DSNU}} \]

Where:

- \( \text{dark\_noise} \) – dark noise

Dark noise is a parameter configurable by the user.

Finally, the Electronic Noise is defined by the equation:

\[ \text{Electronic Noise} = \sigma_{\text{electronic\_noise}} \cdot \text{randn} \]

The Electronic Noise is added to the image in electrons after the DSNU application.

**Electrons to Voltage Conversion**

1. "OC_Factor" is taken as an input.
2. Analogue voltage converted from number of electrons for each detector element is calculated as:

\[ I_{ijk}^V = I_{ijk}^e \cdot \text{OC\_Factor}_k \]

Where:

- \( \text{OC\_Factor}_k \) – is the Output Conversion Factor for the \( k \)-th band

**Voltage Amplification**

1. "Electronic Gain" is taken as an input.
2. CCD voltage is amplified using electronic gain:

\[ I_{ijk}' = G_k \cdot I_{ijk}^V \]

Where:

- \( G_k \) – is the electronic gain for \( k \)-th band
Signal Digitalisation

1. "Max input voltage" and "Bit depth" are taken as inputs.
2. Maximum digitalisation value is calculated as:
   \[ D_{\text{max}} = 2^{\text{Bit Depth}} - 1 \]
   
   Where:
   - \( D_{\text{max}} \) - is the maximum digitalisation value
   - Bit Depth is the number of bits available to store the information (a measure of the radiometric precision)
3. The digitalisation factor is calculated as:
   \[ D_{\text{Fact,}k} = \frac{D_{\text{max}}}{V_{I_{\text{max,}k}}} \]
   
   Where:
   - \( D_{\text{Fact,}k} \) - is the k-th band digitalisation factor
   - \( D_{\text{max}} \) - is the maximum digitalisation value
   - \( V_{I_{\text{max,}k}} \) - is the maximum input voltage for k-th band
4. Detector element data in digital numbers is calculated as:
   \[ I_{ijk}^{DN} = \text{round} \left( I_{ijk}^{\prime} \cdot D_{\text{Fact,}k} \right) \]
   
   The output is the image in digital numbers.

Defective Pixels

Defective pixels represent the severe malfunction of detector elements. There are two main types, when the pixel is non-operating ("dead pixels") and the readout is 0. The second type is the opposite, when the pixel give a constant high value ("hot pixels").

1. The number of dead pixels and hot pixels in the detector line is read from configuration file
2. The dead and hot pixels are distributed uniformly along the detector line.
3. The output of the dead pixel is assigned a 0, the output of the hot pixel is assigned the maximum digital number value. This is modified in all the pixel acquisition, i.e. a whole column of the image.
8.2.2. FLOW DIAGRAM

Figure 8-11 Radiometric block flow diagram
8.2.3. INTERFACES

8.2.3.1. Configuration Parameters

The following configuration parameters are defined.

- Global includes parameters that affect several modules
- Local refers to parameters that affect only this module
- Variable Type is the C data type, from [RD.10]

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
<th>Global or Local</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flag_Photonic_Noise</td>
<td>Flag to activate the photonic noise error</td>
<td>boolean</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>Flag_DSNU</td>
<td>Flag to activate the DSNU error</td>
<td>boolean</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>Flag_PRNU</td>
<td>Flag to activate the PRNU error</td>
<td>boolean</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>Flag_Defective_Pixels</td>
<td>Flag to activate the defective pixels</td>
<td>boolean</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>Min_photon_for_photonic_noise</td>
<td>The Photonic noise is modelled with a Gaussian distribution. The photonic noise is a Poisson process and for large values it follows the Gaussian distribution. For small values of photons this error is not applied. This configuration parameters specifies the minimum value of photons to apply this effect. Photons are typically large values so 10 photons is a reasonable default value.</td>
<td>int</td>
<td>Number of photons</td>
<td>Local</td>
</tr>
<tr>
<td>central wavelength</td>
<td>Wavelength of the centre of the spectral band</td>
<td>Double</td>
<td>m</td>
<td>Global</td>
</tr>
<tr>
<td>fill_factor_x</td>
<td>filled fraction of detector element in X direction</td>
<td>Double</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>fill_factor_y</td>
<td>filled fraction of detector element in Y direction</td>
<td>Double</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>pixel_size</td>
<td>Size of each detector element</td>
<td>Double</td>
<td>m</td>
<td>Global</td>
</tr>
<tr>
<td>Full Well Capacity</td>
<td>Maximum value of charge the detector element can hold before saturating.</td>
<td>Integer</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>Quantum efficiency</td>
<td>Proportion of photons converted to electrons</td>
<td>Double</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>OC_Factor</td>
<td>Analogue voltage converted from number of electrons for each detector element</td>
<td>Double</td>
<td>V/e-</td>
<td>Local</td>
</tr>
<tr>
<td>Electronic Gain</td>
<td>CCD voltage is amplified using electronic gain</td>
<td>Double</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>Max input voltage</td>
<td>Maximum voltage of the scene.</td>
<td>Double</td>
<td>V</td>
<td>Local</td>
</tr>
<tr>
<td>Bit depth</td>
<td>Number of bits in which each value of radiance converted to DN is saved.</td>
<td>Integer</td>
<td>Bit</td>
<td>Global</td>
</tr>
<tr>
<td>Dead pixels</td>
<td>Number of pixels that give a readout of 0</td>
<td>Integer</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>Hot pixels</td>
<td>Number of pixels that give a readout of a high value constantly.</td>
<td>Integer</td>
<td>Dimensionless</td>
<td>Local</td>
</tr>
<tr>
<td>DSNU mean</td>
<td>Dark Signal for the working temperature of the acquisition (considered constant for all the acquisition time).</td>
<td>Double</td>
<td>e-</td>
<td>Local</td>
</tr>
</tbody>
</table>
Name | Description | Variable Type | Units | Global or Local
---|---|---|---|---
DSNU sigma | The variance of the DSNU error | Double | Dimensionless | Local
Dark noise mean | The mean of the dark noise error | Double | Dimensionless | Local
Dark noise sigma | The variance of the dark noise error | Double | Dimensionless | Local
PRNU mean | The mean gain of the PRNU error | Double | Dimensionless | Local
PRNU sigma | The variance of the gain of the PRNU error | Double | Dimensionless | Local

### 8.2.3.2. Inputs

**Table 8-5 Inputs definition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image in irradiance</td>
<td>The input to the radiometric block is the image in irradiances. It is already spatially resampled and includes spatial and spectral aberrations.</td>
<td>Double array [MxN] in NetCDF</td>
<td>[W m⁻²]</td>
</tr>
</tbody>
</table>

### 8.2.3.3. Outputs

**Table 8-6 Outputs definition**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Variable Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image in Digital Numbers</td>
<td>The output of the radiometric block is the binary image in digital numbers.</td>
<td>Integer array [NxN] in NetCDF</td>
<td>Dimensionless</td>
</tr>
</tbody>
</table>

### 8.2.4. SCOPE AND LIMITATIONS

This block is applicable to Earth Observation missions, including missions with the following instruments:

- Passive Opticals
- Active Opticals
- Passive Microwaves
- Active Microwaves

The error model shall be revised, but the radiometric transformation is applicable to

- Passive Opticals
- Passive Microwaves
- Active Microwaves

### 8.2.5. EXTERNAL LIBRARIES

The libraries used inside this block are :

- Eigen, §11.1
- NetCDF §11.5
- EOCFIs §11.6
- Boost §11.4
- Logger §11.3
8.2.6. ERRORS AND WARNINGS.

Exception handling is done with the EOCFI CfiError Error class, [RD.26]. Warning and Errors are reported in the Logger, with a message that allows the user to understand the problem. An exception is thrown in case of error and the execution stops. Warnings do not stop the execution.

For the Radiometric block the following errors can occur:

- File access: input files not found.
- Errors inside the EOCFI functions.
9. ANNEX. SUMMARY OF UNITS

A summary of the units of the inputs and products of BIBLOS is provided here:

<table>
<thead>
<tr>
<th>Product</th>
<th>Units</th>
<th>Block where it is read/calculated</th>
</tr>
</thead>
<tbody>
<tr>
<td>State vector position</td>
<td>m</td>
<td>Blocks in the Geometry module</td>
</tr>
<tr>
<td>State vector velocity</td>
<td>m/s</td>
<td>Blocks in the Geometry module</td>
</tr>
<tr>
<td>Quaternions</td>
<td>Dimensionless</td>
<td>Blocks in the Geometry module</td>
</tr>
<tr>
<td>Observation angles (OZA, SZA, RAA)</td>
<td>rad</td>
<td>Output of the Scene interaction block. Input to the Scene Generation Module.</td>
</tr>
<tr>
<td>Geodetic coordinates</td>
<td>[deg,deg,m]</td>
<td>Output of the Scene interaction block. Input to the Scene Generation Module.</td>
</tr>
<tr>
<td>Reflectance</td>
<td>Dimensionless, multiplied by 1e-4. Values of reflectance are [0-1] They are multiplied by 1e4 and saved in int16 in the range [0-10000]</td>
<td>Input to Atmospheric block</td>
</tr>
<tr>
<td>Radiance</td>
<td>mW/(m2 sr nm)</td>
<td>Output of the Atmospheric block. Input to the Spatial block</td>
</tr>
<tr>
<td>Irradiance</td>
<td>mW/(m2 sr)</td>
<td>Output to the Spatial block. Input to the Radiometric block</td>
</tr>
<tr>
<td>Digital counts</td>
<td>Dimensionless. Values go from 0-2^bit depth-1.</td>
<td>Output to the Radiometric block</td>
</tr>
</tbody>
</table>
10. ANNEX. EOCFI LIBRARY.

10.1. LIST OF CLASSES AND METHODS USED

For information on the specification of each class of the EOCFI library refer to [RD.26].

Table 10-1 List of EOCFI Classes used and use in BIBLOS.

<table>
<thead>
<tr>
<th>EOCFI Class</th>
<th>Use in BIBLOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CfError</td>
<td>Exception handling</td>
</tr>
<tr>
<td>ModelId</td>
<td>Model of the ellipsoid. This is a global parameters used across BIBLOS. The current version (v4.9) does not allow to change the ellipsoid properties.</td>
</tr>
<tr>
<td>TimeCorrelation</td>
<td>Initialisation of the time changes. Initilised reading the global parameters with XLCFI_TIMEMOD_IERS_B_PREDICTED mode of the EOCFI, and the IERS file defined in the global configuration file.</td>
</tr>
<tr>
<td>OrbitId</td>
<td>Use of the osvCompute method for the orbit propagation in the orbit block.</td>
</tr>
<tr>
<td>FixedHeader</td>
<td>Class to print header to the Orbit and Attitude files in EOCFI format. Used when writing Attitude and Orbit into EOCFI format.</td>
</tr>
<tr>
<td>XmlFile</td>
<td>EOCFI class that handles the XMLs. Used when writing Attitude and Orbit into EOCFI format.</td>
</tr>
<tr>
<td>OrbitFile</td>
<td>Orbit file handling class. Used when writing Orbit into EOCFI format.</td>
</tr>
<tr>
<td>OsvRec</td>
<td>Class to store the Orbit in EOCFI format. Used when writing Orbit into EOCFI format. To work with the state vector the StateVector class is used from EOCFI.</td>
</tr>
<tr>
<td>AttFile</td>
<td>Attitude file handling class. Used when writing Attitude into EOCFI format. To work with the attitude the Eigen library is used for its mathematical capabilities (conversion to quaternions, and matrix, vector operations).</td>
</tr>
<tr>
<td>AttRec</td>
<td>Class to store the attitude (or other rotations) in EOCFI format. Used when writing Attitude into EOCFI format.</td>
</tr>
<tr>
<td>StateVector</td>
<td>Class that contains the state vector and time.. Used across BIBLOS to do operations such as ECI to ECEF conversion, and to store and work with state vectors. When doing mathematical operations the StateVector is converted to an Eigen vector.</td>
</tr>
<tr>
<td>Geodetic</td>
<td>Class used across BIBLOS to work with geodetic coordinates and to convert ECEF to geodetic.</td>
</tr>
<tr>
<td>DemId</td>
<td>Initialised while reading the global config file according to the configuration file defined there. The access to DEM is done using the 'compute' function. Used in the Scene Interaction Block.</td>
</tr>
</tbody>
</table>

10.2. DIFFERENCES WITH RESPECT TO EOCFI

In this section some differences with respect to the EOCFI library are described. There is no impact to the simulation chain.

10.2.1. ORBITAL FRAME

What is defined as the Orbital Frame in EOCFI is not equivalent to the definition in BIBLOS. The definition according to EOCFI is shown here, the definition from BIBLOS is defined in §13.7. There Orbital Reference frame is used internally but all products (state vectors) are always given in ECEF of ECI.
10.2.2. INTERSECTION WITH ELLIPSOID

The intersection with the Ellipsoid in the EOCFI Libraries is done in the Pointing
In the Pointing Library of the EOCFIs there are a number of methods of the Target class, [RD.26].

From [RD.26]:
"Target::targetInter: It calculates the intersection point(s) of the line of sight defined by an elevation
and an azimuth angle expressed in the input Attitude frame, with a surface(s) located at a certain
geodetic altitude(s) over the Earth."

In BIBLOS the intersection with the ellipsoid at an altitude is done with the position of the pixel and
line of sight in ECEF, as described in §Intersection with the Ellipsoid.
11. ANNEX. EXTERNAL LIBRARIES USED IN BIBLOS

11.1. EIGEN

All Vector, Matrix and Rotation Matrix operations are done with the Eigen library. Eigen is a consistent, reliable and user-friendly library.

11.2. KISS FFT

The Kiss FFT Library is used for discrete direct and inverse Fourier transforms in the Spatial Block.
Reference: Kiss FFT, <kissfft.sourceforge.net/>

11.3. EHLOGGER

The EHLogger Library is part of the distribution of the OpenSF package, [RD.27]. BIBLOS uses this library to print to console the messages.
The EHLogger does not print to a log file, so if the user wants to save the log to file it must do so either by executing through OpenSF (as OpenSF saves the log file), or by adding a ‘tee’ command in the execution line.

11.4. BOOST

Boost library is used for vector, matrix operations, managing files, file system package and unit tests.
Reference: Boost <http://www.boost.org/>

11.5. NETCDF

NetCDF is used to read and write products.
Reference: NetCDF <http://www.unidata.ucar.edu/software/netcdf/>

11.6. EOCFIS

The EO CFIs are used for a number of operations including operations of orbit, attitude, geolocations, exception management, etc, [RD.26]
Reference: EO CFIs <http://eop-cfi.esa.int/>
12. ANNEX. SPECTRAL RESOLUTION FOR THE LUTS

The stepsize used to generate the LibRadTran LUT should be chosen with reference to two aspects:
- the spectral bands and bandwidths most commonly used in the most popular commercial satellite sensors.
- the amount of radiative energy transmitted by the atmosphere in the given spectral range

Taking into account that most popular satellite remote sensing sensors acquire imagery in the visible and near infrared regions (Aggarwal), and that most research is based on information obtained in these bands (Sahoo, 2011; Smith, 2012), it is well justified to implement a 1nm stepsize in these regions of the electromagnetic spectrum.

Certain substances present in the atmosphere, such as water vapour, carbon dioxide and ozone, absorb solar radiation in specific bands (i.e. atmospheric windows) of the electromagnetic spectrum limiting the amount energy which can be registered by sensors. The atmospheric windows are as follows (Lusch et al., 1999) (Ali, 2010):
- visible and near infrared range - very high transmission (several small H\textsubscript{2}O absorption features between 0,8-1,1µm)
  - 0,3-1,3µm
- middle infrared - high transmission but several very strong H\textsubscript{2}O and CO\textsubscript{2} absorption bands at 1,4; 1,9 and 2,7µm
  - 1,5-1,8µm
  - 2,0-2,6µm
- thermal infrared - frequently used thermal windows
  - 3,5-5,0µm
  - 8-14µm

When modelling the atmosphere, these bands must of course be taken into account. When generating LibRadTran LUT’s, the bandwidth of the atmospheric window in the analysed spectral range will directly determine the minimum stepsize which should be used. Using a 50nm step size in the middle infrared range will only give 6 LUT values in the first atmospheric window (1,5-1,8µm) and only 12 values in the 2,0-2,6µm window. Such a small about on data points will usually be insufficient for my types of analyses. However if a 10nm step size is used, this gives 30 LUT values in the first mentioned window and 60 measurements in the second. The LUTs will have a 10nm stepsize in the 1,5-2,6µm range.
When analysing the thermal infrared region, the use of such narrow bandwidths will yet again generate a very large number of LUT values. **Therefore, using the same assumptions as above, the LUTs will have a 50nm stepsize in the 3,5-14 µm range**, giving 30 measurements in the 3,5-5,0µm range and 120 data points in the 8-14µm range.

**References:**

D. Lusch, W.D.Hudson, "Introduction to environmental remote sensing", 1999


Shefali Aggarwal, "PRINCIPLES OF REMOTE SENSING"

R.N. Sahoo, "Hyperspectral remote sensing"

R.B. Smith, "TNTmips - Introduction to hyperspectral imaging", 2012
13. ANNEX. REFERENCE FRAMES AND TRANSFORMATIONS

13.1. ECEF

ECEF frame utilized throughout the project (unless specifically stated otherwise) is International Terrestrial Reference Frame (ITRF) as recommended by IUGG [RD.20] further referenced as ‘ECEF Frame’ or ‘ITRF’. ITRF frame is defined as:

- Origin of the system is in the barycenter of Earth (with all landmass, oceans and atmosphere)
- Z-axis of FK5 frame is normal to the Earth’s mean equatorial plane of J2000.0 Epoch.
- X-axis is defined by the intersection of the Earth’s mean equatorial plane of J2000.0 and Earth’s prime meridian (Greenwich Meridian) or more precisely by Terrestial Intermediate Origin (TIO)
- Y-axis completes the right-hand system

For detailed definition refer to [RD.21]

Within BIBLOS ECEF is taken as the Earth Fixed frame (XLCFI_CS_EF) defined in the EOCFIIs Conventions document, [RD.28]

13.2. ECI

ECI frame utilized throughout the project (unless specifically stated otherwise) is Geocentric Celestial Reference Frame (GCRF) as defined by IAU [RD.18] further referenced as ‘ECI Frame’ or ‘GCRF’.

One of the recommendations of GCRF definition was that it should be as close as possible to, previously used in astronomical community, FK5 system. FK5 frame is defined as:

- Origin of the system is in the barycenter of Earth
- Z-axis of FK5 frame is normal to the Earth’s mean equatorial plane of J2000.0 Epoch.
- X-axis is defined by the intersection of the Earth’s mean equatorial plane of J2000.0 and mean Ecliptic plane of J2000.0 and it points towards Aries, or more precisely – it is defined by Celestial Intermediate Origin (CIO) [RD.18],[RD.19].
- Y-axis completes the right-hand system

For detailed definition refer to [RD.18].

Within BIBLOS ECI is taken as the Geocentric Mean of 2000 (XLCFI_CS_GM2000) defined in the EOCFIIs Conventions document, [RD.28]

13.3. ECEF <-> ECI TRANSFORMATION

Conversion between ECI AND ECEF frames, in particular between GCRS and ITRS are thoroughly described in IERS Technical Note No. 36, chapter 5 [RD.21] and as such will not be discussed here. The implementation is also available in Standards of Fundamental Astronomy (SOFA) software libraries [RD.11]

The transformation is done with the EOCFI libraries, see §9. and [RD.28]

13.4. GEODETIC REFERENCE FRAME

The geodetic reference frame is a coordinate system that defines a position based on the Earth ellipsoid. The position is defined by 3 parameters:

- Latitude – north-south position on the Earth surface
- Longitude – east-east position on the Earth surface
- Altitude – vertical direction with respect to the local horizon from the ellipsoid.
13.5. ECEF TO GEODETIC TRANSFORMATION

Given the definition of the ellipsoid with the flattening, semi-major axis, semi-minor axis and eccentricity. And given the definition of a point in ECEF \((x, y, z)\). The calculation of the geodetic coordinates \((\text{lat}, \text{lon}, \text{alt})\) is done with the EOCFI libraries, see §9.

13.6. GEODETIC TO ECEF TRANSFORMATION

Given the definition of the ellipsoid with the flattening, semi-major axis, semi-minor axis and eccentricity. And given the definition of a point in geodetic coordinates \((\text{lat}, \text{lon}, \text{alt})\). The calculation of the ECEF coordinates \((x, y, z)\) is done with the EOCFI libraries, see §9. and [RD.26].

13.7. RESTITUTED ORBITAL REFERENCE FRAME

The Orbital Frame as used in BIBLOS is defined by:

- Origin of the system is in the Restituted centre of mass of the spacecraft
- Z-axis of spacecraft is in the direction of geocentric nadir (Z-axis cuts with the centre of the Ellipsoid).
- Y-axis is vector perpendicular to the velocity and the Z-axis. \(\text{Y-axis} = \text{Z-axis} \times \text{u_vel}\)
- X-axis completes the right-hand system. The X-axis is very close to the velocity vector. If the eccentricity is 0, the X-axis and the velocity vector are the same. Given that Earth observation orbits have a small eccentricity, the axis are close but not equal.

Figure 13-1: Orbital Frame

13.8. SATELLITE REFERENCE FRAME

The Satellite Frame is an auxiliary reference frame between the Instrument frame and the Orbital frame. The difference between the orbital and the Satellite frame is the Attitude. The Attitude is the rotation matrix from Satellite to Orbital frame.

For a geodetic with yaw steering attitude mode (default mode), the satellite frame is defined by:
• Origin of the system is in the centre of mass of the spacecraft
• Z-axis of spacecraft is in the direction of geodetic nadir (Z-axis in the direction of the local vertical).
• Y-axis is the vector perpendicular to the velocity corrected with the Earth rotation (for the yaw steering) and the Z-axis.
• X-axis completes the right-hand system

13.9. INSTRUMENT REFERENCE FRAME

The Instrument Frame as used in BIBLOS is defined by:
• Origin of the system is in the centre of the detector line of the spacecraft
• X-axis is in the Along-track direction of the detector line, within the plane of the detector line
• Z-axis is perpendicular to the detector line plane, with +Z-axis in the direction of the Line-of-sight
• Y-axis is in the Across-track direction of the detector line, completing the right-hand system.

![Diagram of Instrument Frame](image)

Figure 13-2: Instrument Frame, X-Y plane (=focal plane)

Line of sight of element \( \gamma \); the +Z in the instrument frame is in the same direction (pointing towards the Earth)

![Diagram of Line of Sight](image)

Figure 13-3: Instrument Frame

13.10. ORBITAL TO ECEF AND ECI TRANSFORMATIONS

Given the orbital vectors X-axis, Y-axis and Z-axis expressed in ECEF, the Orbital to ECEF transformation is defined by:

\[
\text{Orb2ECEF} = [X\text{-axis}, Y\text{-axis}, Z\text{-axis}]
\]

Where X, Y and Z are 3x1 (in columns), and are defined as:
The transformation from Orbital to ECI is the same with the Orbital vectors expressed in ECI.

13.11. EULER ANGLES, ROTATION MATRIX AND QUATERNIONS CONVERSIONS.

The conversion between Euler Angles, rotation matrices and quaternions are done with the EOCFI library. For the reference of the conventions for the Euler Angles and the Quaternions please refer to the EOCFI Conventions SUM, [RD.26].

13.12. UNIVERSAL TRANSVERSE MERCATOR COORDINATE SYSTEM (UTM)

The Transverse Mercator projection is an adaptation of the Mercator projection and is shown in the diagram below. Both projections are cylindrical and conformal. However, in the transverse Mercator, the cylinder is rotated 90° (transverse) relative to the equator so that the projected surface is aligned to a central meridian rather than to the equator, as is the case with the equatorial Mercator projection.

Currently, the WGS84 ellipsoid is used as the underlying model of the Earth in the UTM coordinate system.

The UTM system is not a single map projection. The system instead employs a series of sixty zones covering 6° in longitude, each of which is based on a specifically defined secant transverse Mercator projection as shown below:
A position on the Earth is referenced in the UTM system by the UTM zone, and the easting and northing coordinate pair.

The precise form of the formulas used for the UTM projection is given in [Annoni et al] p. 115-117. Pages 114-115 of [Annoni et al] describe the ETRS Geodetic Datum (ETRS89-TMzn) recommended for Europe. For the rest of the world, the global WGS84 Datum will be used. Essentially, this means that for negative latitudes (i.e. for the Southern Hemisphere), a false northing of 10,000,000 m must be used.

One important consideration when using UTM is the fact that the image may naturally fall between two UTM zones, although the image would necessarily need to be projected into a single zone. Here there would appear to be several logical choices for the zone:

1) The zone corresponding to a corner of the image (e.g. the top left corner).
2) The zone corresponding to the center of the image.
3) The zone where the majority of the image is found.

Alternatively, the product could contain two images projected in the two zones, allowing the user to choose the most suitable image. This however would duplicate image size and is probably undesirable.

The proposal is to apply rule 3) above and take the zone where the majority of the image is found, although the choice could be configurable by the user. The objective should be, in any case, to avoid negative coordinates and possible distortions as far as possible.

---

14. ANNEX. TIME REFERENCES

This section describes the time references. In BIBLOS the time references are handled with the TimeCorrelation class of the EOCFIs, [RD.26] [RD.28].

14.1. MODIFIED JULIAN DAY (MJD2000), EPOCH

The Modified Julian Date 2000 (MJD2000) is the day count from 00:00 on the 1st of January 2000. Times in BIBLOS are given in MJD2000 unless specified otherwise.

The term ‘epoch’ as used in BIBLOS described the time expressed in MJD2000.

14.2. JULIAN DAY (JD)

Julian Day (That is the interval of time in days and fractions of day since the 1st of January 4713 BC Greenwich noon). JD is not used in BIBLOS, and is only defined here for reference.

The relation with MJD2000 is the following:

\[ \text{MJD2000} = \text{JD} - 2451544.5 \]

14.3. UNIVERSAL TIME (UT)

The International Telecommunications Union Recommendation (ITU-R TF.460-6), Standard-frequency and time-signal emission, defines the following:

"Universal time (UT) is the general designation of time scales based on the rotation of the Earth. In applications in which an imprecision of a few hundredths of a second cannot be tolerated, it is necessary to specify the form of UT which should be used:

UT0 is the mean solar time of the prime meridian obtained from direct astronomical observation;

UT1 is UT0 corrected for the effects of small movements of the Earth relative to the axis of rotation (polar variation);

UT2 is UT1 corrected for the effects of a small seasonal fluctuation in the rate of rotation of the Earth;

UT1 is used in this Recommendation, since it corresponds directly with the angular position of the Earth around its axis of diurnal rotation. Concise definitions of the above terms and the concepts involved are available in the publications of the IERS (Paris, France)."

14.4. GREENWICH (GMT)

The Greenwich Mean time is UT1 as defined in the previous section.

14.5. COORDINATED UNIVERSAL TIME (UTC)

The International Telecommunications Union Recommendation (ITU-R TF.460-6), Standard-frequency and time-signal emission, defines the following:

"UTC is the time-scale maintained by the BIPM, with assistance from the IERS, which forms the basis of a coordinated dissemination of standard frequencies and time signals. It corresponds exactly in rate with TAI but differs from it by an integer number of seconds. The UTC scale is adjusted by the insertion or deletion of seconds (positive or negative leapseconds) to ensure approximate agreement with UT1."

The relation with TAI is the following:

\[ \text{TAI} = \text{UTC} + \text{Leap_seconds} \]

14.6. INTERNATIONAL ATOMIC TIME (TAI)

The International Telecommunications Union Recommendation (ITU-R TF.460-6), Standard-frequency and time-signal emission, defines the following:

"The international reference scale of atomic time (TAI), based on the second (SI), as realized on the rotating geoid, is formed by the BIPM on the basis of clock data supplied by cooperating establishments. It is in the form of a continuous scale, e.g. in days, hours, minutes and seconds from the origin 1 January 1958 (adopted by the CGPM 1971)."

The relation with TAI is the following:
TAI= UTC+Leap_seconds

The reference time used in BIBLOS is TAI.

14.7. LEAP SECONDS

The International Telecommunications Union Recommendation (ITU-R TF.460-6), *Standard-frequency and time-signal emission*, defines the following:

"2.1 A positive or negative leap-second should be the last second of a UTC month, but first preference should be given to the end of December and June, and second preference to the end of March and September.

2.2 A positive leap-second begins at 23h 59m 60s and ends at 0h 0m 0s of the first day of the following month. In the case of a negative leap-second, 23h 59m 58s will be followed one second later by 0h 0m 0s of the first day of the following month (see Annex 3).

2.3 The IERS should decide upon and announce the introduction of a leap-second, such an announcement to be made at least eight weeks in advance."

14.8. TIME CONVERSIONS

The time conversions are handled with the EOCFI libraries, see §9. and [RD.28]. The reference time used in BIBLOS is TAI.
15. ANNEX. INTERPOLATION METHOD

The propagation and interpolation used for the state vector is the osvCompute function from the OrbitId class of the EOCFIs.

For the quaternions interpolation is done with the quaternionsInterpol function from the Lib function. For a detailed specification refer to [RD.26].