RUGGED: AN OPEN-SOURCE SENSOR-TO-TERRAIN MAPPING TOOL

L. Maisonobe, A. Espesset, G. Prat

CS-SI
Parc de la grande plaine - 5 rue Brindejonc des Moulinais
BP 15872
31506 Toulouse CEDEX 5
France

ABSTRACT

The Sentinel-2 mission is a component of the Copernicus program. It consists of two spacecrafts each carrying a high resolution multispectral imager (13 bands) devoted to environmental, security, and agricultural applications. One of the key feature of Sentinel-2 is the huge amount of imagery data that will be produced, as each spacecraft produces 1.7TB of raw data daily. All data are transformed on ground at the Image Processing Facility to create the various products levels.

One element of this processing chain is the Rugged library: an open-source library built on top of the open-source Orekit space flight dynamics library. Rugged is used to compute very quickly and accurately the mapping between ground points and on-board pixels, taking into account a Digital Elevation Model. Direct location computation is used to identify which ground point is seen by a specified sensor pixel. Inverse location computation is used to identify which sensor pixel will see a specified ground point. These methods are the basic elements for complete processing algorithms. They are computationally intensive due to the very large number of pixels to manage (12 detectors, 13 bands, global coverage of land surfaces).

The geo-location methods are at the boundary between image processing and flight dynamics as they handle accurate geometrical models. Rugged has therefore been designed as an intermediate level library, and it relies on Orekit to compute the global geometry (spacecraft orbit and attitude, Earth precession nutation and proper rotation including all IERS corrections, Earth mean ellipsoidal shape). The Rugged library adds on top of this the Digital Elevation Model intersection computation.

The presentation will describe the overall Rugged technical architecture, the issues that were faced and how they were solved in order to achieve a high performance level while not compromising accuracy. It will also present the open-source strategy and the governance model of the library. The library is already operationally used, some perspectives for it will be discussed.

Index Terms— Open-source, Rugged, Orekit, image processing, geolocation, ortho-rectification, viewing model, line-of-sight, digital elevation model

1. INTRODUCTION

Looking at raw data acquired by an optical Earth Observation (EO) satellite can be rather disappointing. Color may not be as uniform as one would expect, the images may lack sharpness, and the geometry of the scene is strongly distorted. It is impossible to overlay different spectral channels and reconstitute a color image, nor to map the image on Earth. That is why the acquisition raw data are never disseminated to the end-users without pre-processing at the user ground segment. The first level of end-user products are obtained following a series of radiometric and geometric corrections which aim at removing the artefacts induced by the instrument and by the viewing geometry. Level-1 products are generated systematically from the raw data at the Instrument Processing Facilities (IPF) of the user ground segment - or Payload Data Ground Segment (PDGS) which is in charge of delivering high-quality products to the end-users.

In this paper we will focus on the geometric corrections of the IPF. Level-1 geometric processing aims at correcting the distortions of the image which have multiple causes: the wide viewing angle of the instrument, the spheroidal shape of the Earth, the distance between the satellite and the ground, the rotation of the Earth, the motion of the satellite, and the relief of the Earths terrain. This is where the open-source software Rugged plays a key role. This library allows to model complex viewing geometry with instruments comprising multiple detectors and viewing angles. It can calculate the intersection of the line-of-sight of each individual pixel with the Earths rugged terrain at any time, hence determining which point on the ground each pixel is looking at.

Ruggeds algorithms are extremely precise and take into account all the physics of the system: Earth rotation, satellite motion and attitude, instrument orientation (with the possibility of defining the orientation by a series of rotations with respect to the platform), light travel time, etc. Most of the cal-

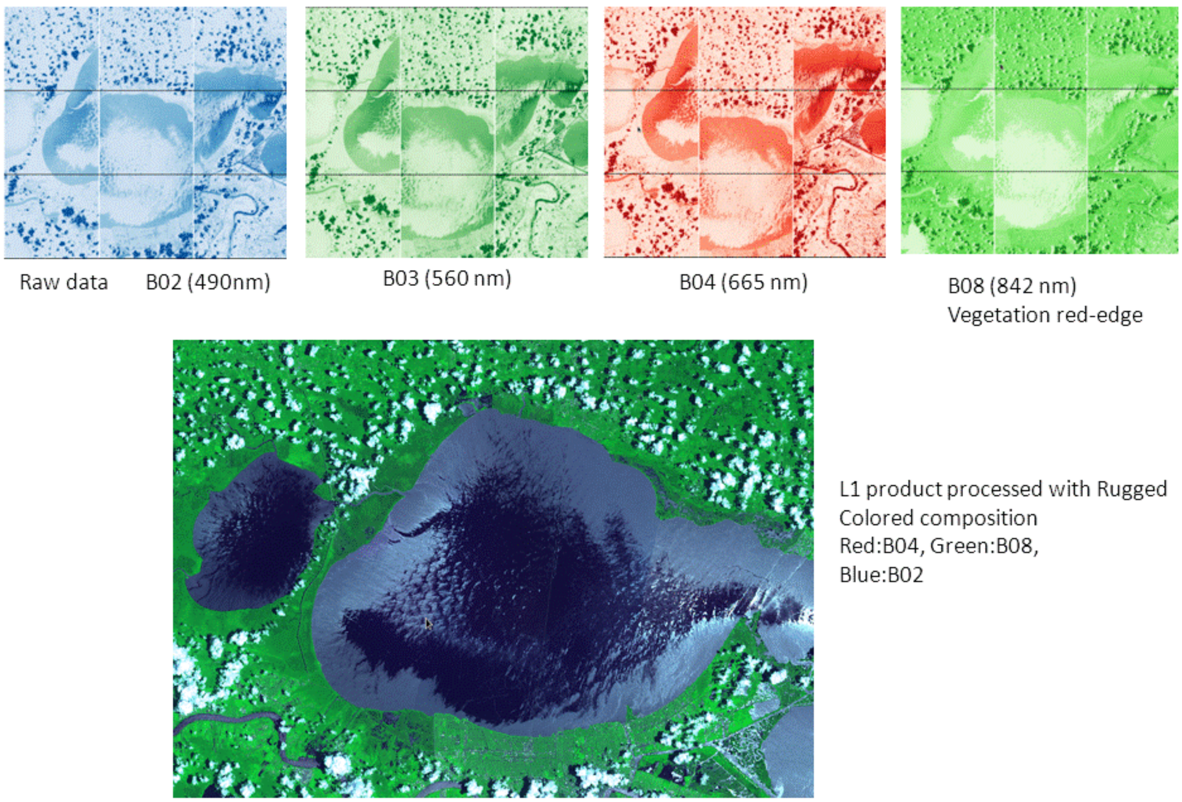

Fig. 1. Sentinel-2 acquisition of Lake Pontchartrain, Louisiana, Copernicus Sentinel data [2015]

culations related to the physics of the Earth and of the satellite are available through the underlying open-source library named Orekit (Orbit Extrapolation kit), a well-known flight-dynamics software.

What Rugged brings on top of Orekit, is the capability to take into account a Digital Elevation Model (DEM) in the computation of the line-of-sight intersection with the Earth. Rugged offers an interface dedicated to performing direct (focal plane to ground) and inverse (ground to focal plane) locations of single points or of multiple points. The latter are handled via the generation of grids of points, called mapping grids in this article; they can be the results of direct or inverse location functions or colocations (direct then inverse in order to map one focal plane onto another).

Rugged is an intermediate-level library which provides all the interfaces and functions to model any type of sensor but does not provide any specific model. It is up to the caller to instantiate a specific sensor and to define its characteristics.

Released in January 2015, Rugged is already operational at the Sentinel-2 ground segment since S2-A first acquisitions which occurred in June 2015. The first products were processed with success on the same day; the commissioning phase has started and should last until fall 2015. S2 mission is very demanding in terms of performance and robustness as a consequence of the extremely large amount of data that will be acquired and will need to be processed. The twin satellites S2-A and S2-B will acquire high resolution images continuously over land in 13 spectral bands, with a 290km swath. The entire Earth will be covered in 5 days leading to 3 Terabytes of data per day. The main challenge of the ground segment lies on its capacity to process, archive and disseminate the end-user products in a timely manner; in consequence strong requirements were imposed on the design and performance of all underlying core libraries such as Rugged.

Section 2 focuses on the functionalities of Rugged, while section 3 describes how Rugged was instantiated for Sentinel-2 and put to good (and operational) use at the ground segment.

Figure 1 shows a Sentinel-2 image processed with Rugged. First row shows raw data of 4 single channels, second row is a color composition at level-1.

2. RUGGED

2.1. Design

Rugged has been designed as an intermediate layer library strictly focused on direct and inverse location computation, that could be reused for different missions and contexts. These characteristics have driven a few fundamental design choices:

- Rugged would only provide features related to direct and inverse location,
- Rugged would not include any mission-specific parts, they must be implemented in higher layers
- Rugged would not include low level frame transforms and should rely on lower level validated libraries for these computations
- Rugged would not handle images by itself, it would only create the mapping grids
- Rugged would not load Digital Elevation Models by itself, they should be loaded by higher layers and passed through

These choices have led to the layers shown in Figure 2.

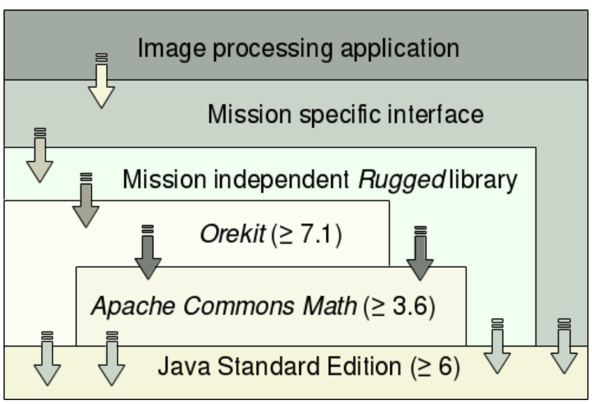

Fig. 2. Rugged layered architecture

The existing and validated Orekit [1, 2, 3] lower level open-source [4, 5] library provides space flight dynamics computation, including accurate frames transforms (inertial, terrestrial) taking into account all Earth Orientation Parameters [6] provided by IERS, as well as ellipsoid or geoid Earth models, time handling, orbit and attitude propagation, and interpolation among other things. The Apache Commons Math open-source [7] library provides the core mathematical algorithms like geometry core packages, root solvers, optimizers or ODE integrators.

On top of these existing libraries, two layers have been added: one, mission independent Rugged open-source [4] library; the other, a mission-specific layer. This mission specific layer typically handles the spacecraft viewing model, the time-stamping models, as well as loading of the Digital Elevation Model selected for the mission and loading of all orbit, attitude and calibration metadata. Adapting Rugged to a new mission therefore requires changing only a well-controlled and precisely defined part, without changing the core mission-independent core. No image processing is done at this level; the interface between this layer and the applications above corresponds to mapping grids.

The top level layer corresponds to the various image processing applications that will use the mapping grids in order to fully process the image themselves to perform ortho-rectification, calibration, registration...

Top level applications will typically have their own set of dependencies (for example GDAL [8] to load geospatial data like Digital Elevation Models or Orfeo ToolBox [9] to process large images), some of which may be huge. The simple layered architecture adopted for Rugged brings a minimum number of new dependencies and fits well within an existing framework without any incompatibility. It is therefore easy to integrate in any processing chain.

2.2. Features

As Rugged is designed to be used in any type of Earth observation missions, it needs to take care of all possible geometric effects that influence the mapping between ground and on-board sensor. The ground points moves as the Earth rotates with respect to the inertial frame. Accurate models of this motion are already available in Orekit and validated at millimeter level. One legacy models still widely used in Earth observation is the old Lieske precession, Wahr nutation and mean sidereal time, they are collectively known as IAU 1980 models and were the reference models up to IERS conventions 19968. These models are available and all the Earth Orientation Parameters (EOP) corrections that need to be applied on these old models are transparently loaded and applied directly by Orekit. The EOP corrections include the UT1 time correction on proper rotation, the LOD correction on the rotation derivative, the xp and yp pole motion (polhody) and the $\delta\Delta\psi$ and $\delta\Delta\varepsilon$ corrections on nutation. When all corrections are properly applied, very high accuracy can be reached. In addition to these legacy models, more modern ones are also available like the IAU-2000 nutation, IAU-2006 precession computed from the new Non-Rotating Origin paradigm introduced with IERS 2003 [11] and IERS 2010 [6] conventions rather than the equinox-based paradigm that prevailed before. Here again, a set of Earth Orientation Parameters corrections are automatically applied. Of course, the corrections applied to the newer models are much smaller than the corrections applied to IAU-1980 models since the models are more accurate. With these models, centimeter level accuracy is reached. In fact, it is also possible to ask Orekit for even greater accuracy by enabling the use of tidal corrections on the EOP interpolation, but this is only required for sub-centimeter accuracy, which is far below Digital Elevation Model accuracy and as such irrelevant for this kind of applications. One important note is that these very accurate frames are available simply by selecting them from a list, there is no need for the users to know at all what these models are, how to implement them and how to apply corrections. Everything is done transparently and users simply have to drop the EOP files to some known folder when they are published by IERS. Rugged therefore adapts to both legacy systems and modern systems and is ready for the most accurate needs.

The elevation of ground points is modeled using a raster digital elevation model. In order to allow different missions to select different models (many missions use a tailored DEM sometimes with specific formats), the model is not loaded in the mission-independent Rugged layer but in the mission-specific layer, and passed to Rugged after loading. For efficiency reasons, the elevations passed to Rugged are expected to be relative to the ellipsoid, but the DEM loaded could be defined with respect to the geoid if the mission-specific layer takes care of the geoid undulation before passing the data. The geoid undulation can be extracted from another raster file like the EGM-96 geoid, but Orekit can also help here as it can compute a geoid from any gravity field (Orekit knows how to load gravity fields in several standard formats). The mainly vertical displacement of ground points due to solid tides is not considered yet as digital elevations models uncertainty is often much greater than this displacement.

The path of the line of sight from sensor to ground point is evaluated with respect to the Digital Elevation Model taking into account the ellipsoid curvatures, thus allowing mapping even for missions with either agile spacecraft that can reach large deflection angles or sensors with very large fields of view.

For validation purposes and in order to compare output with existing products that approximated the final part of the line of sight as a straight line in geographic coordinates (thus enabling simple Cartesian interpolations between DEM raster cells), it is possible to disable the full computation and enable a flat Earth flag. The error of ignoring the curvature is null at nadir (90° incidence) and increases rapidly as the incidence angle decreases. With a 40° angle, the interpolation error due to enabling the flat Earth flag already reaches 4.5m and increases dramatically with smaller angles. Due to this large error, it is strongly recommended to disable the flat Earth flag which was only intended for comparison. The full computation overhead is only 3% with respect to the simplified model.

The free space travel light time is compensated for each pixel independently. This effect distorts the image as the ground points are not at the same place at the time they emit light and at the time this light is received on board. Compensating for this effect means taking into account Earth orientation change during the free space travel. The absolute value of this correction amounts to about 1.2m for a Low Earth orbiting spacecraft. Compensating this effect independently for each pixel allows accurate computation even for wide scans where the points at nadir are closer to the spacecraft than the points away from nadir. The differential effect across the field of view is very small (a few centimeters), but allows Rugged to be ready for very demanding missions too. If needed, this correction can be switched off, mainly for validation purposes.

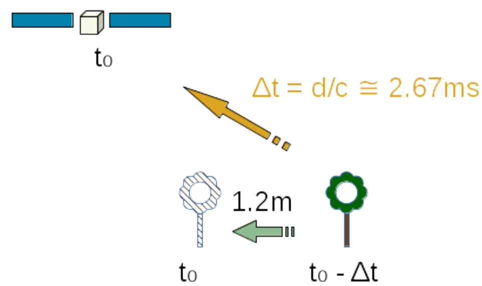

Fig. 3. Light time correction

Apart from the light time correction, the aberration of light is also compensated. This phenomenon is due to the spacecraft velocity and not ground point velocity as light time correction. Both phenomena are similar from a physical point of view, and they depend simply on the frame in which the computation is performed. For Rugged, everything is computed in an inertial frame and both ground and spacecraft move with respect to this frame, so both phenomena should be computed. The absolute value of this correction amounts to about 20m for a Low Earth orbiting spacecraft. This correction can be also switched off, but it is not for validation purposes: it may be required if the viewing directions in spacecraft frame already take this effect into account, hence, it should not be counted twice.

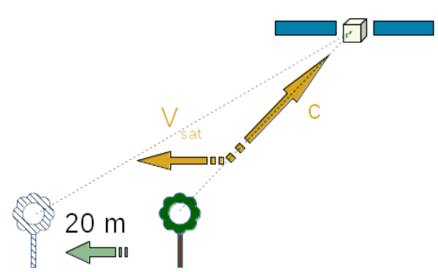

Fig. 4. Aberration of light correction

On board, the directions of each pixels line of sight are used to compute the mapping, and these directions can be time-dependent with any user-provided model. A typical case is to use a low degree polynomial for some of the angles that define the spacecraft viewing model, as a sequence of transforms from the sensor chip to instrument frame to spacecraft frame. These low degree polynomials use time as their independent variable and model the effects of thermo-elastic transforms as the spacecraft moves from eclipse to daylight and back to eclipse.

One correction that is still missing in the current Rugged version is refraction effects. These effects are of the order of magnitude of 1-2m, mainly for viewing directions far from nadir. This was not needed for Sentinel-2 as the push-broom sensor is centered on nadir and 10° wide and as the tolerance was larger than this effect. It is however a limitation for more demanding missions so it is planned to add it. Several approaches are possible and in fact we will certainly propose an interface with a Rugged provided reference implementation and a possibility for the users to plug their own models tailored for their mission.

2.3. Performance

As the Sentinel-2 mission needs to process a very large amount of data, it was important to meet some stringent performance requirements. Rugged is now considered to be very efficient. This efficiency was achieved first because the underlying Orekit library performs rapid and optimized frame transforms, even considering the large IAU-2000/2006 precession nutation model (several thousand Poisson series terms). Two different levels of caches with interpolation and Taylor expansions are used to speed up the computation. Rugged also pre-computes many combined transforms out of the innermost pixels loops. This involved some intricate computation as light-time correction (and hence Earth orientation) should be adjusted independently for each pixel, but it proved to be feasible and to work very well.

Another mean to achieve good performances was to use a very fast algorithm for Digital elevation Model intersection. We selected the Duvenhage [12] algorithm. This algorithm allows to quickly pruning out parts of the line of sight known not to intersect the DEM. The min/max k-dimensional tree used in the model is created automatically at DEM load time at Rugged level. It is done in linear time and linear space. This algorithm is used for direct location.

The inverse location function has also been designed in a non-classical way to be as fast as possible. The classical approach is to use a bi-dimensional optimizer (along-track/across-track) with calls to the direct location function to check the convergence criteria iteratively until convergence. In Rugged, the bi-dimensional optimizer has been replaced by two fast mono-dimensional solvers (along-track, then across-track) followed by a refinement step without any call to the time-consuming direct location function. All these computations take advantage of Orekits ability to compute analytical derivatives of frames transforms which are at the heart of the inverse location mapping.

With all these enhancements, the current speed is the following (tests done on a 2 years old Dell XPS notebook with a quad core Intel i5-3320M CPU at 2.6Ghz and 8Gio RAM, running GNU/Linux Debian and using openJDK8): the direct location computes about 85000 pixel/seconds in mono-thread and the inverse location computes about 65000 pixel/seconds in mono-thread.

2.4. Open source and governance

The Rugged library is freely available as free software both in source and binary formats, with all related documentation and tests. It is published under the terms of the Apache V2 license [4], which belongs to the category of permissive licenses. This means anybody can use it to build any application, free or not.

The governance adopted for the project is similar to the governance adopted for the Orekit project, and is based on the Apache meritocratic model. According to this model, people get more power as they gain merit working for the project. The entry level corresponds to users, which are not even known by the project. They are people who anonymously downloaded the product and use it without notifying anyone. The next level corresponds to contributors, who identified themselves with the project and brought some contributions, in the form of either code, documentation, bug reports, test cases, data, discussions on the mailing lists or help to other users. Once contributors have made a number of contributions showing a real involvement in the project, they may be elected to the next level, which corresponds to committers. Committers have write access to the project repository so they can change the code. Being elected committer implies signing a Contributor License Agreement (either an Individual Contributor License Agreement or a Corporate Contributor License Agreement depending on the conditions). These agreements are the legal contracts that allow the Rugged team to publish the committed code. The last level in the meritocracy corresponds to PMC members, for the Project Management Committee which is the steering board of the project, in charge of roadmap and project administrative lead (including voting in new committers and new PMC members). Rugged governance is open, meaning anyone can be elected at any level of the meritocracy independently of their origin.

3. SENTINEL-2 GETS RUGGED

3.1. Overview of Sentinel-2 mission

- two satellites S2A (launched in June 2015) and S2B (launch planned for 2016)
- systematic coverage of all land and coastal areas between 84°N and 56°S every 5 days (10-day revisit time per satellite)

- multi-spectral instrument with 13 spectral bands in the VNIR and SWIR:
  - 4 spectral bands at 10m,
  - 6 at 20m,
  - 3 at 60m ground resolution
- 290 km swath
- End-user Level-1 products are Top-Of-Atmosphere (TOA) reflectance projected in UTM coordinates and provided as 100km^2 tiles according to the UTM MGRS scheme.

3.2. Sentinel-2 viewing model

Fig. 5. Schematic view of one VNIR detector

Sentinel-2 Multi-Spectral Instrument (MSI) is a push-broom optical imager built with a Three-Mirror Anastigmat (TMA) telescope and a dichroic beam splitter which directs the light on two distinct focal planes for the VNIR and SWIR channels. Both focal planes comprise 12 detectors in order to cover the 290km swath required. Separation of the individual spectral channels is achieved using filters overlaid on top of the detectors. The 12 detectors are placed in a staggered alignment in order to avoid any gap between the images. The figure below shows the configuration of the VNIR focal plane and illustrates the complexity of the geometry. The design of the instrument, driven by the requirements on high resolution, high radiometric precision and large swath, leads to misalignments between spectral bands, detectors and focal planes that need to be overcome during Level-1 processing.

Figure 5 is a schematic view of one VNIR detector. Colored lines illustrate the spectral channels with double lines for TDI (time-delay integration) detectors (B03 and B04). The number of pixels and their size depend on the desired resolution (2596 pixels per detector for a 10m band).

Figure 6 shows the detectors alignment in the focal plane. The staggered configuration induces inter-detector parallax angles with a minimum value for B02 and a maximum for B09 reaching 46km shift along-track. The inter-band parallax angle is maximum between B02 and B09 and results in a 14km shift along-track.

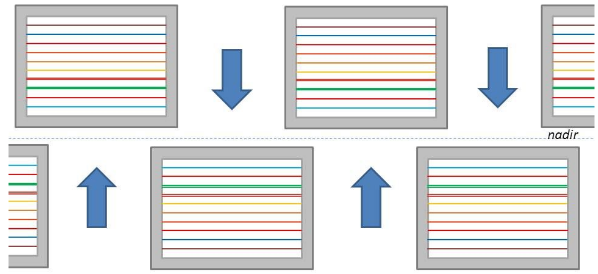

Fig. 6. Detectors alignment in the focal plane

3.3. Sentinel-2 product levels

L0C consolidated level-0 product : image data are raw data in on-board compressed format. Satellite auxiliary data have been decommutated and associated to the image data

L1A product : decompressed raw data for visualization (focal plane)

L1B product : after applying radiometric corrections (focal plane)

L1C product (end-user) : ortho-rectified Top-Of-Atmosphere reflectance values in UTM coordinates

From L0C to L1B, products are divided in granules of approximately 23 by 25km containing the 13 spectral bands. L1C products are divided in UTM MGRS tiles of about 110km2.

3.4. Interfaces with Rugged

The Sentinel 2 processing chain has been designed as a set of Instrument Data Processing Software Components (IDP-SC) in order to generate L0 to L1C products. Each IDP-SC performs one or several functions at a given processing step. For example IDP-SC QL_GEO performs the preliminary QuickLook resampling function at Preliminary Quick look processing processing step. The IDP-SC components are implemented in C++. They use the Orfeo Toolbox library (OTB) framework for radiometry processing while relying on Rugged for their geometrical needs. The mission-specific library s2geo implements Sentinel-2 viewing model and acts as the layer between Rugged and the IDP-SC components. S2geo performs a set of specific tasks for the IDP-SC. Implemented in Java like Rugged, it is called from the C++ IDP-SC components via file-based interfaces. The main interface between the IDP-SC and s2geo specifies the tasks to be executed

Level	Task	s2geo/Rugged macro-functions
Level 0C	Level-0 Viewing Model Initialization	Granule footprint computation Global footprint Solar and viewing angles
Level 0C	Preliminary Quick-Look Processing	Colocation grid
Level 1A	Level-1 Viewing Model Initialization	Granule footprint Global footprint Detector footprint Solar and viewing angles
Level 1A	Cloud Mask re-projection	Mask projection
Level 1C	Tile metadata	Grid of solar angles Grid of incidence angles
Level 1C	Resampling Grid Computation	Inverse location grid
Level 1C	Mask Projection	Mask projection

Table 1. S2geo and Rugged roles in Sentinel-2 level-1 processing chain Level

and contains all the processing parameters, including paths to dedicated parameter files (viewing direction files, Digital Elevation Model, etc.). Each call to Rugged is a macro-function (for instance compute a mapping grid). Once Rugged has completed its steps, the IDP-SC processes the outputs. This pipeline is illustrated on Figure 7.

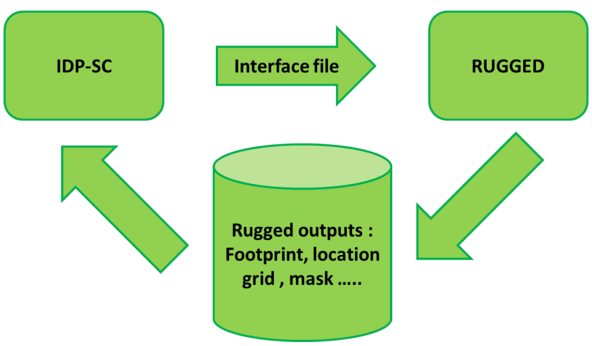

Fig. 7. Interfaces between IDP-SC, s2geo and Rugged

Table 1 below lists the functions performed by s2geo for the IDP-SC. These are macro-functions which rely on Rugged at the low level.

3.5. Validation

The geometrical accuracy has been validated by comparison with reference data produced by the Ground Processor Prototype (GPP). The GPP has been developed jointly by ESA/CNES based on long-standing CNES proprietary image processing tools, qualified on many high-resolution missions. The GPP is used during the commissioning phase by the Sentinel-2 Technical Expertise Centre. In terms of geometrical performance, the project requirement for L1C cross-product geolocation was 0.3 pixels at 3σ compared to the GPP reference data (i.e. 3 meters for 10m resolution bands). This was validated before acceptance based on simulated data, and confirmed on real acquisitions during the commissioning phase during which a number of products were compared with the GPP for final validation of the operational chain.

3.6. Ortho-rectification performance

Rugged performance is discussed in chapter 2 where the processing times for generating direct and inverse grids are given. Here, we complete this information with the performance achieved for Sentinel-2 image ortho-rectification. The time budget for this task is 15 minutes to process one UTM tile (about 110km2) in mono-thread, which amounts to 670 Mpixels. The ortho-rectification component includes the following steps:

resampling grid generation : inverse location grids are produced by s2geo/rugged with a parametrized step (benchmark executed with 45m for 10m bands, 90m for 20m bands, and 180m for 60m bands for performance assessment)

resampling : location grids are used with a parametrized interpolator (SPLINE5 for performance assessment), to produce a 110x110km tile in UTM coordinates. This step is done using the library OTB7.

reflectance conversion : image radiance is converted to top-of-atmosphere reflectance.

Performance benchmarking has been performed on an Intel Bi Xeon i5 2640 at 2.5 GHz and 32GO ECC RAM in mono-thread. Results are summarized in Table 2. The overall speed is 0.7Mpixels per second, i.e. 10 million pixels are processed in 14 seconds.

TASK	Time (in seconds)
s2geo / Rugged (generation of the resampling grid)	**427**
Initialization (IO, viewing model initialization, etc.)	9
Resampling grid	418
for one 45m resolution grid	73
for one 90m resolution grid	18
for one 180m resolution grid	5
IDP-SC (resampling and reflectance conversion)	**418**
Other tasks	24
Resampling + Reflectance conversion	394
for one 10m band	290
for one 20m band	95
for one 60m band	9
Total	**845**

Table 2. Benchmark processing time for ortho-rectification (mono-thread)

4. CONCLUSION

Rugged, a new-born open-source library for precise geolocation, created by CS-SI, is at the heart of Sentinel-2 operational processing at ESA (European Space Agency). Designed and implemented to fit any type of sensors, it is waiting for new challenging missions. Rugged is eager to build a larger community: anyone is welcome to go to the site www.orekit.org/rugged , download Rugged and use it, participate to the mailing lists, contribute and later become a committer or even a PMC member.

If you carpet the Earth, make sure its Rugged!

5. REFERENCES

[1] L. Maisonobe, V. Pommier-Maurussane, Orekit: an Open-source Library for Operational Flight Dynamics Applications, in 4^{th} *ICATT*, May 2010.

[2] L. Maisonobe, P. Parraud, From low level toolbox to orbit determination: handling users requests in Orekit in 6^{th} *ICATT*, Mar 2016.

[3] Orekit site: `https://www.orekit.org/`.

[4] Orekit Apache Software License V2, `https://www.orekit.org/rugged/license.html`.

[5] L. Maisonobe, P. Parraud, S. Dinot open-source publication: a strategic choice for private companies, in 6^{th} *ICATT*, Mar 2016.

[6] G. Petit, B. Luzum, IERS Conventions (2010), IERS Technical Note 36, 43-71, (2010)

[7] Apache Commons Math: `https://commons.apache.org/math/`

[8] Open Source Geospatial Foundation, GDAL website, `http://www.gdal.org/`

[9] CNES and OTB contributors, Orfeo ToolBox, `https://www.orfeo-toolbox.org/`

[10] D. D. McCarty, IERS conventions (1996), IERS Technical Note 21), 20-39, (1996)

[11] D. D. Mc Carty, G. Petit, IERS Conventions (2003), IERS Technical Note 32, 33-56, (2003)

[12] B. Duvenhage B., Using An Implicit Min/Max KD-Tree for Doing Efficient Terrain Line of Sight Calculations, in 6^{th} International Conference on Virtual Reality, Computer Graphics, Visualisation and Interaction in Africa., Centurion, South Africa, 2009