

NWP SAF AAPP VIIRS-CrIS Mapping

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 29 June 2011, between EUMETSAT and the Met Office, UK, by one or more partners within the NWP SAF. The partners in the NWP SAF are the Met Office, ECMWF, KNMI and Météo France.

Copyright 2014, EUMETSAT, All Rights Reserved.

Change record			
Version	Date	Author / changed by	Remarks
1.0	20.11.14	P. Brunel	Initial Document version

CONTENTS

REFERENCE DOCUMENTS.....	2
1 INTRODUCTION.....	3
2 CRIS FOOTPRINT.....	3
3 MAPPING TO VIIRS.....	5
3.1 BOW-TIE EFFECT.....	5
3.2 VIIRS ALONG SCAN SPATIAL RESOLUTION.....	6
3.3 ADJACENCY AND NEIGHBOURHOOD.....	7
4 IMPLEMENTATION OF CRIS-VIIRS MAPPING IN AAPP.....	10
4.1 LIST OF FILES.....	11
4.2 CALLING GRAPH FOR MAIN PROGRAM.....	11
5 RESULTS AND VALIDATION.....	13
5.2 TEST ON FULL ORBIT.....	14
5.3 MAIA4 RESULTS.....	17
6 CONCLUSION AND FUTURE WORK.....	18
ANNEX AAPP L1C L1D VARIABLES UPDATED BY MAPPING.....	19

Reference documents

Ref 1: Cross Track Infrared Sounder (CrIS) Sensor Data Records (SDR) Algorithm theoretical Basis Document (ATBD) Code 474-00032 July 31, 2011

Ref 2: VIIRS Geolocation Algorithm Theoretical Basis Document (ATBD) code 474-00053 July 31, 2011

Ref 3: Document for Common Adjacency Software, Operational Algorithm Description (OAD) code 474-00097 June 3rd, 2013

1 Introduction

This document presents the VIIRS-CrIS mapping software. The Cross-track Infrared Sounder (CrIS) on board Suomi-NPP has a ground footprint of 14km at nadir. A good knowledge of the surface conditions will benefit any use of CrIS radiances. The Visible Infrared Imaging Radiometer Suite (VIIRS) can provide exclusive information about clouds or surface in the CrIS field of view. The CrIS AAPP level 1c and 1d formats has some reserved variables for VIIRS data. The AAPP command `viirs_to_cris` will feed those variables with relevant VIIRS data.

This document explains the method, the command and gives some results. The software description document is separate and has been automatically generated by Doxygen tool.

This command override any use of navigation routines that needs instrument geometry, but use all geo-location data provided with the SDR files. So one will use an off-line reference data base of CrIS footprints for all fields of regard, all field of view, one scan line, one satellite altitude. The most important thing to take care is the particular VIIRS scanning characteristics which implies that different pixels may have the same geographical position and that the next pixel may not be the true next neighbour. One will use the "adjacency" method (see ref 1) which is able to provide the true nearest VIIRS neighbours for a given VIIRS pixel.

2 CrIS footprint

The AAPP navigation routines are been used to calculate viewing vectors for the CrIS optical axis and centre field of views. Some routines already exist in AAPP for the IASI scanning, they were modified according the CrIS scanning mode which is a bit different. The field of regard contains 9 field of views and rotates by 45 degrees along the scan (see Figure 1 and Figure 2).

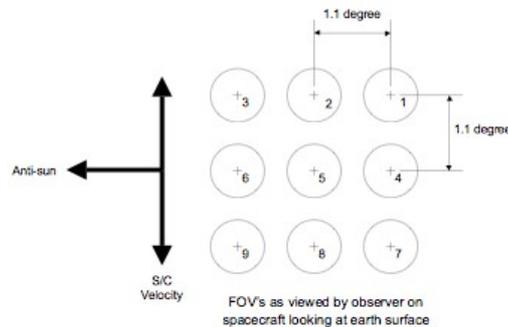
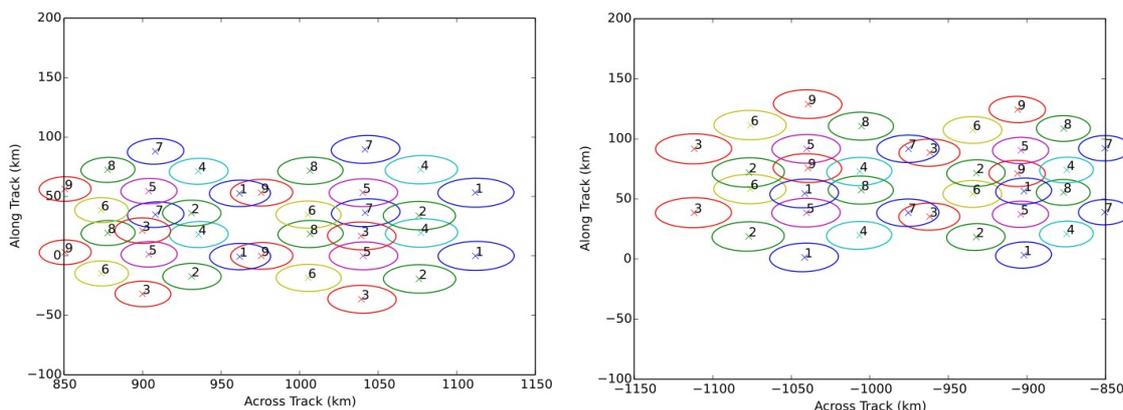


Figure 1: CrIS field of regard definition (from Ref1)



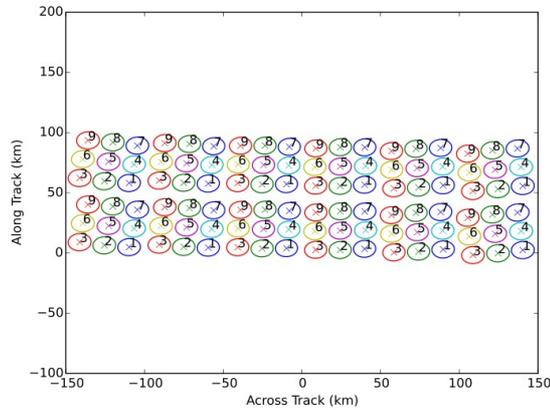


Figure 2: CrIS footprint on a perfect plane for two scans. The FOR rotates by 45 degrees along the scan. Top left, FOR 1 and 2, Top right FOR 29 and 30, Bottom center FORs.

The CrIS footprint is considered as a cone (16.8mrad) centred on the centre field of views. The intersection with Earth is converted in across and along track distances. The size of the footprint varies from 14x14km at nadir to up to 48x24km on the edges. The geolocation of each FOV is validated against the CrIS SDR data, results give a very good agreement. The accuracy of the exact geolocation is not a goal, we need to ensure that the scanning equations and routines are properly coded. The important result is the size of the field of view, and this does not need a very good accuracy of the geolocation. This is the only place where navigation software has been used in the study.

For each field of view the four semi-axes are calculated for several full orbit passes. For a given FOV the footprint size is a function of scanning angles and satellite altitude. Given a reference altitude, it can be considered that the axis size is linear with the satellite altitude difference. Complete calculations are done for a set of orbits and regression coefficients are derived. The two regression coefficients are stored in the fixed data file mapping_viirs_cris.fdf.

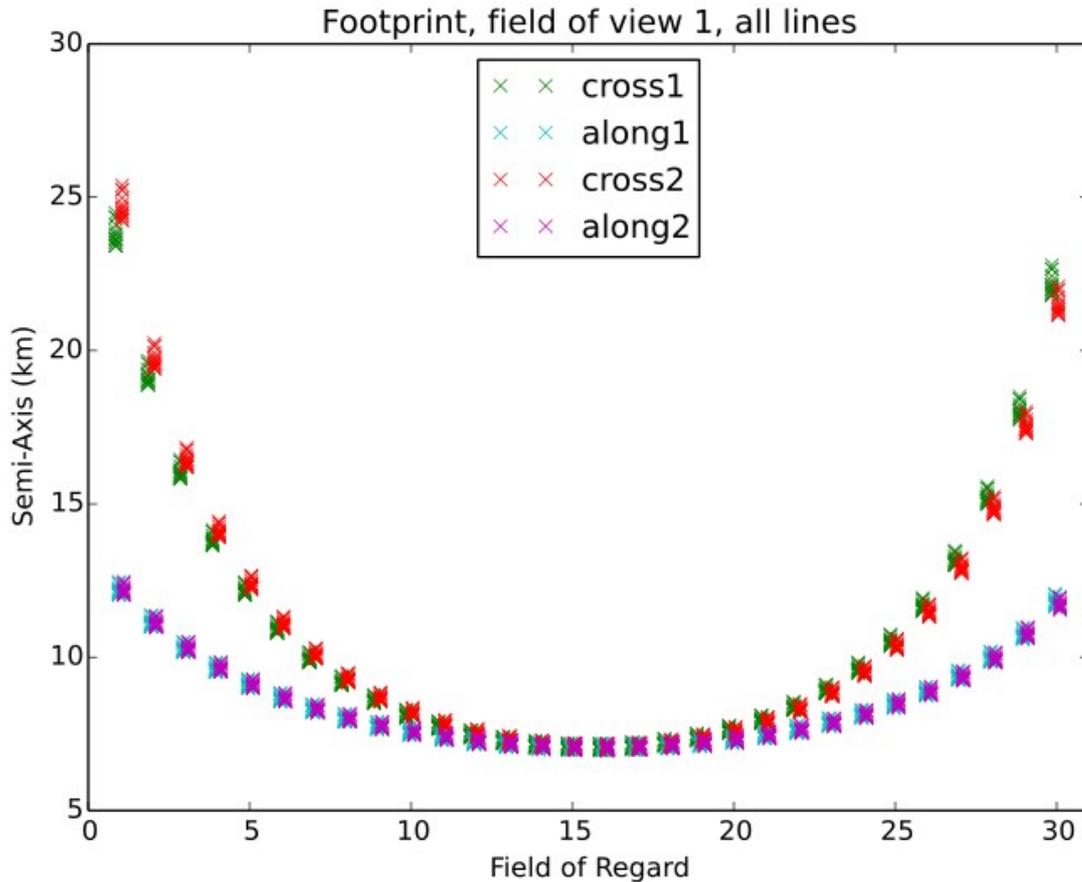


Figure 3: Footprint axis size for field of view number 1, all scan lines of a typical orbit.

3 Mapping to VIIRS

3.1 Bow-Tie effect

The bow-tie effect (Ref 2) of the VIIRS instrument complicates the mapping of the CrIS field of view (see Figure 4 and Figure 5). We use a Fortran90 generalisation of the Common Adjacency algorithm (Ref 3) to find the exact VIIRS pixels that fall into the CrIS FOV. The original software only considers three VIIRS scans which limits the search radius to 16 pixels/lines. Our version of source code has been extended without software limits, so we can use it for the CrIS mapping, or anything else.

Any user who wants to average some VIIRS data inside a perimeter should take care of the bow-tie effect. Inside a VIIRS scan there is no issue, all neighbours are real neighbours, but if one try to cross scans the neighbourhood is not straightforward and the true next pixel can be part of the next scan (Figure 5). This is managed by the “adjacency” algorithm. It calculates the distance between adjacent rows and columns. One try to find the true pixel next to the considered one, the algorithm selects the pixel which is at a distance of one nominal pixel. This algorithm should be used in any case when a user needs to check local variability or needs to do some statistics.

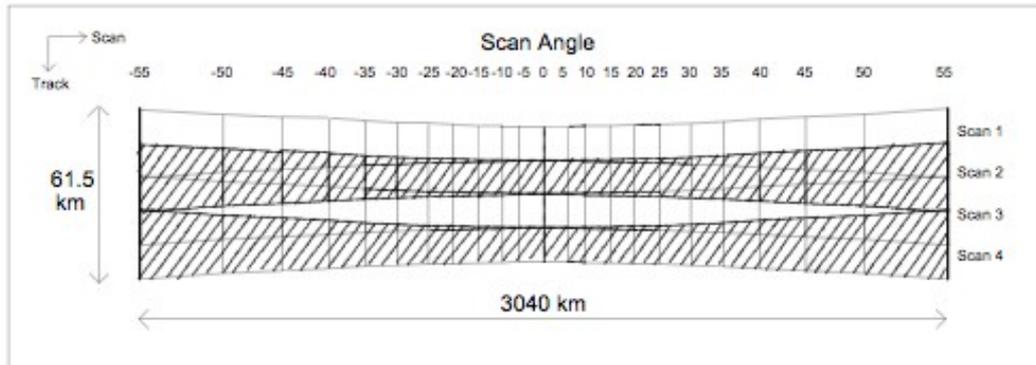


Figure 4: Panoramic bow tie effect (from Ref2)

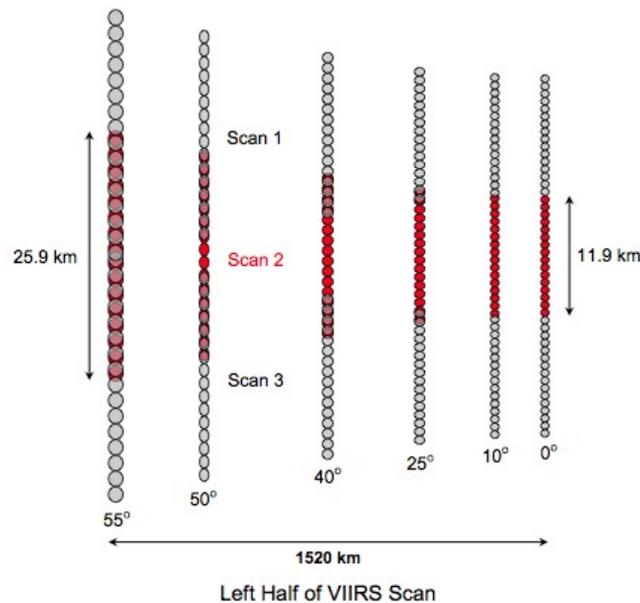


Figure 5: Pixel growth and overlap within a scan (from Ref2). We can see that the 3 scans overlap at the end of the scan; some ground geolocation can be seen 3 times.

3.2 VIIRS along scan spatial resolution

The VIIRS scanning is continuous and regular. The scan is divided in 5 zones where pixels are aggregated with a factor of [3,1,1,2,3] (Figure 6). The aggregation provide a more uniform ground sample size and moderates the pixel growth so the spatial size for band M is in the range [750, 1600] metres.

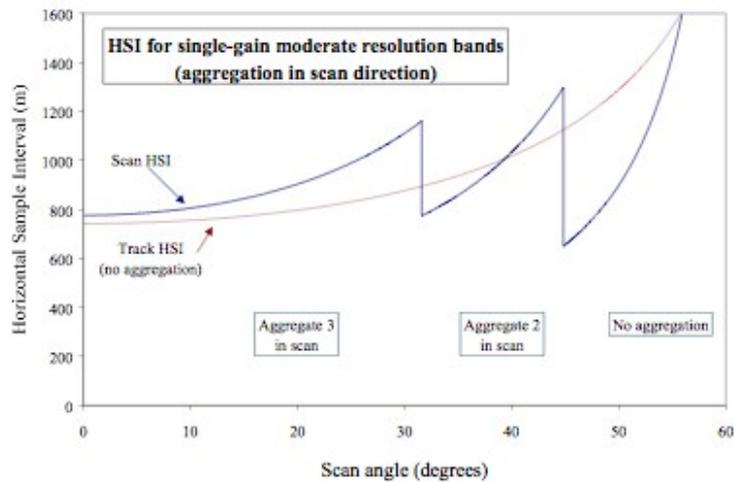


Figure 6: Growth of spatial element ground field of view

When a CrIS field of view is overlapping an area where the aggregation factor is changing, the projected CrIS FOV on the VIIRS image needs a specific treatment.

3.3 Adjacency and neighbourhood

The size of a CrIS FOV is so large that it may cover 3 VIIRS scans. The adjacency algorithm has been coded in fortran 90 and extended to cover any number of scans. The figures Figure 7 and Figure 8 present the impact of the VIIRS scanning on neighbourhood. If we plot the latitude values for an array of VIIRS pixels we can clearly see discontinuities. Moreover when we plot great circle distance of pixels with the centre of the box array, it is evident that closer pixels are not always the closest in distance. The adjacency algorithm solves this issue.

Note that the adjacency algorithm is only able to manage non terrain corrected geolocation.

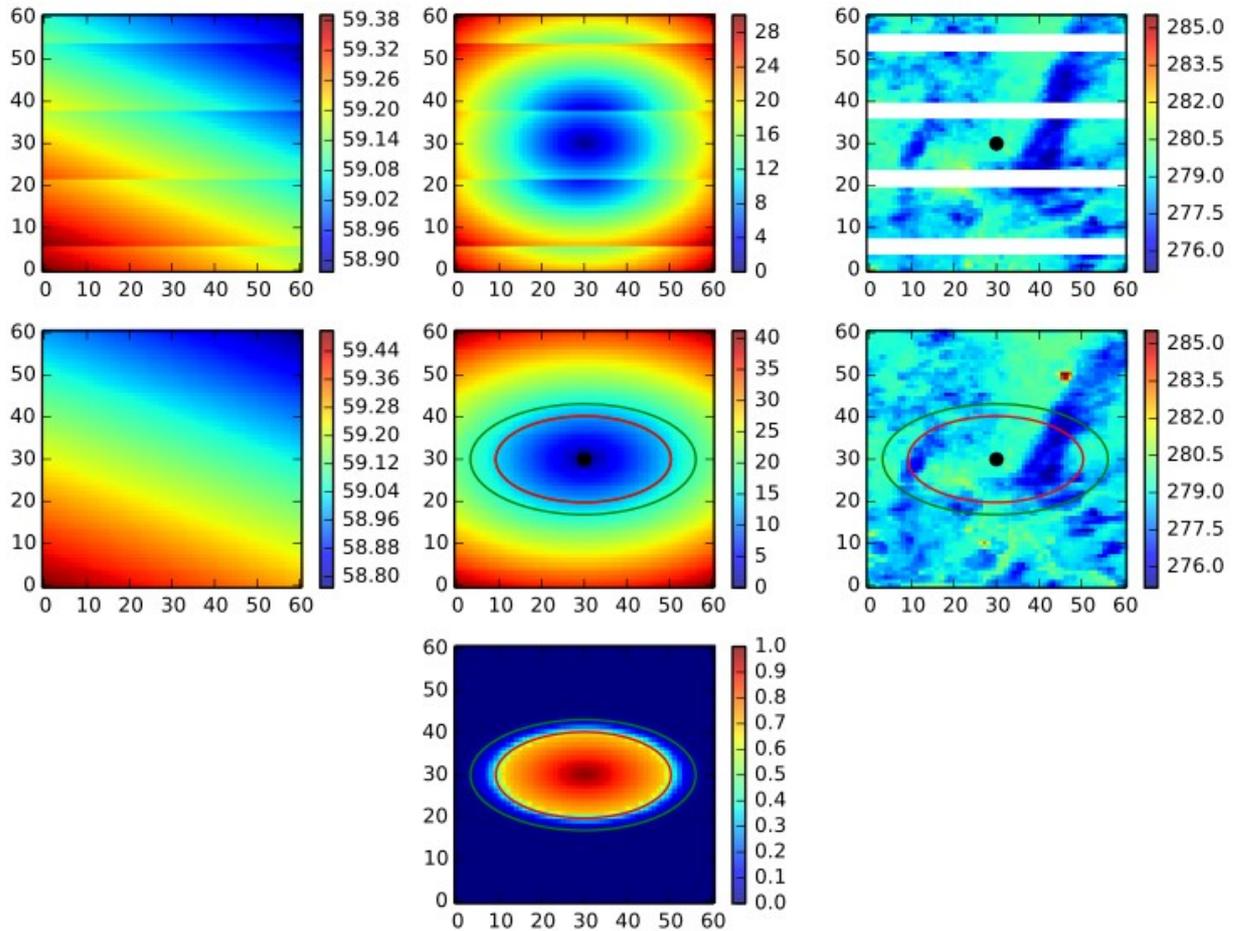


Figure 7: The figure illustrates the VIIRS scanning and a simulated CrIS field of view mapped to VIIRS image. The box is centred on the VIIRS pixel 2600 (see black circle) with a fictitious ellipse of 12x14 km. The top row shows the raw data as read from the VIIRS files. The centre row shows the same data after the adjacency algorithm software has been applied. Left column, plot of the latitudes of the VIIRS pixels. Centre column, plot of the distance in kilometres from centre box. Right column, plot of the VIIRS brightness temperature, see missing values for the overlapping pixels. Bottom image shows a simulated CrIS FOV shape projected on the VIIRS image after reconstruction.

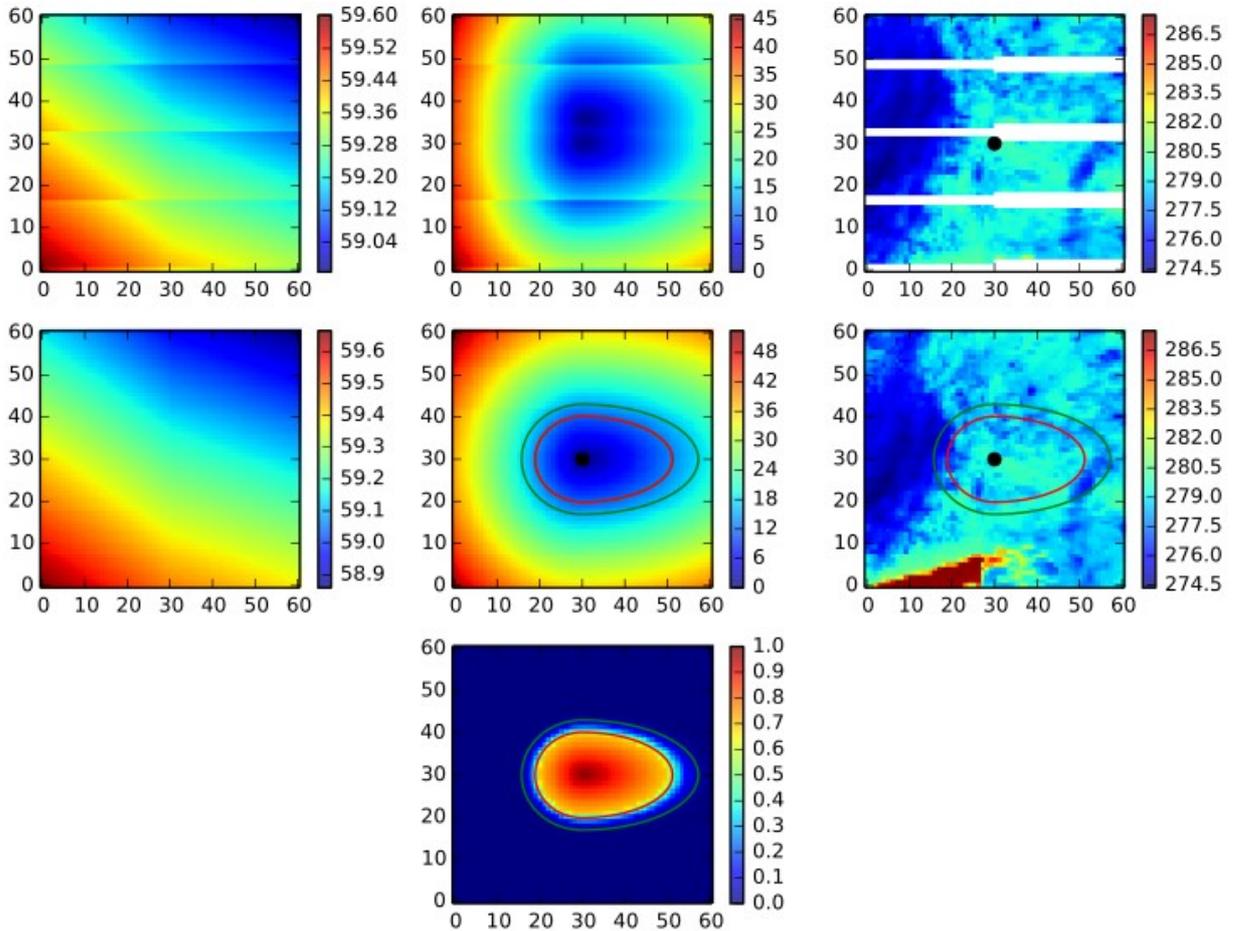


Figure 8: Same as Figure 7, for an extreme case. The centre FOV lies on a bow-tie border pixel and at a VIIRS resolution changing step, i.e. pixel 2560, line 413.

We can clearly see with Figure 8 that the algorithm does a good job in taking the right pixels and setting a good FOV shape. The CrIS ellipse is shown in red and green colours for two different FOV enlarging coefficients (1 and 1.2). The projected CrIS ellipse is highly transformed, this justifies the use of two different semi-axis for mapping. The bottom row presents the weights that can be applied to the VIIRS convolution following the nominal FOV shape characteristics.

	FOV Shape (degrees, Cross Track)	FOV Shape (degrees, In Track)	FOV Matching Band-to-Band, In-track and Cross-track (degrees)
70% of Peak Response Width	> 0.8735	> 0.8735	+/- 0.0206
50% of Peak Response Width	0.942	0.942	+/- 0.0137
10% of Peak Response Width	< 1.100	< 1.100	+/- 0.0206
3% of Peak Response Width	< 1.238	< 1.238	N/A

Figure 9: CrIS FOV shape characteristics

<p>The EUMETSAT Network of Satellite Application Facilities</p>		<h1>VIIRS-CrIS mapping</h1>	<p>Doc ID : NWPSAF-MF-UD-011 Version : 1.0 Date : 20.11.14</p>
---	---	-----------------------------	--

4 Implementation of CrIS-VIIRS mapping in AAPP

The Adjacency algorithm has been coded in Fortran90 and is stored in a module named `map_viirs_adjacency`. It calculates the nearest VIIRS pixels of a given one, taking care of the bow-tie pixel trim, and Earth rotation between scans. This step takes some time to run, as for all VIIRS pixels there are some trigonometric calculations. One has optimised a routine that is used a lot of time for distance calculation between successive pixels. The optimisation makes two assumptions, the Earth is flat and the conversion of longitude difference to kilometres is done with a tabulated cosine of latitude rounded to $1/100^\circ$.

Another Fortran90 module is created for the mapping, named `map_viirs_cris_mod`. It is in charge of setting the FOV size, searching for the nearest VIIRS pixel to the CrIS FOV centre (2 to 4 iterations), building the "true" neighbours and calculating the FOV nominal ellipse and weights relative to FOV shape. A Fortran90 "mapping" structure is returned to user level containing all necessary information to be able to map user data. Two convolution subroutines are provided, one for VIIRS SDRs, and one for MAIA4 cloud mask. Those routines are subject to be modified by the user, so one could retrieve any kind of product.

Note that VIIRS radiances are converted from $W/m^2.sr.\mu m$ to $mW/m^2.sr.cm^{-1}$ for compatibility with CrIS in 1d file, the reference wavelength used for the conversion are stored in the fixed data file.

A main program and an associated script `viirs_to_cris` are provided. The program only takes VIIRS I or M channels at once, because the adjacency table is relative to the band. All available channels in the SDR are considered for the convolution. The user can run the program several times with the same CrIS 1d file, new results will not overwrite previous ones, so one can first run band I and then the band M.

CrIS input file can be AAPP level 1c or level 1d and may contain any number of CrIS granules. Level 1d files basically store only one field of view (see `atovpp` command), if user wants the mapping to consider all FOVs one should update the spatial thinning in fixed data file `data/preproc/CRIS.fdf` and put 0 for full resolution.

A Fortran 77 file `cris1c1d.F` is in charge of the interface between Fortran 90 code and the input/output AAPP routines which are coded in Fortran 77.

VIIRS SDR files (and optionally MAIA4) should be aggregated before calling the mapping in order to cover the CrIS data. One recommends the use of the `nagg` software (<http://www.hdfgroup.org/projects/jpss/>).

VIIRS geolocation file can be provided separately but the default is to use the geolocation data stored in the SDR file. Geolocation and SDR should be entirely compatible. When VIIRS geolocation is terrain corrected (GITCO or GMTCO) the mapping un-corrects any position above 50m high to an altitude lower than 10m by an iterative approximate method.

The FOV shape (Figure 9) can be considered during the convolution step by the convolution routines. The convolution of the VIIRS data takes care of the FOV shape for VIIRS radiances, but this is not considered for the mapping of cloud mask data.

CrIS 1c files are only updated if MAIA4 data is available, as the 1c format only provides space for VIIRS cloud products. CrIS 1d files are updated for VIIRS radiances and reflectances and cloud mask; see annex for details.

The user level command is:

```
viirs_to_cris [-d|-D] [-t threshold] [-b band] [-m Maia4file]
              [-g Geofile] CrISfile VIIRSfile

where band      is a VIIRS band name I or M
Maia4file      is a VIIRS MAIA 4 HDF5 file
Geofile        is a VIIRS geolocation HDF5 file
Crisfile       is a CrIS AAPP level 1c/1d file
VIIRSfile      is a VIIRS SDR HDF5 file
-d debug level 1
-D debug level 2
threshold      is the minimum percentage of valid VIIRS pixels for
mapping
```

If debug is set, log and debug files are created in the working directory.

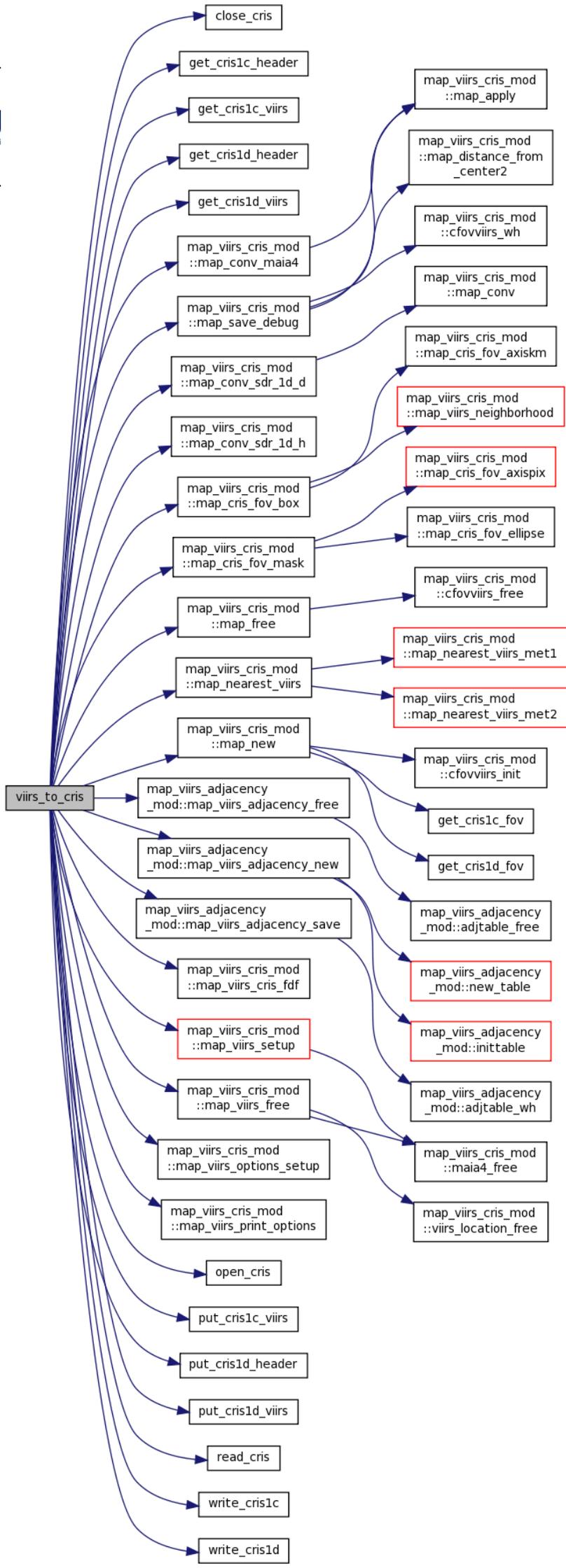
The software modules are commented with doxygen marks. So the full software documentation has been generated by doxygen and is available as a pdf separate document and as html pages.

4.1 List of files

Source code files update to VIIRS-CrIS mapping are

```
src/maia4/
  bin
    viirs_to_cris.F90
    viirs_to_cris.ksh
  libmapviirscris
    map_viirs_adjacency_def.F90
    map_viirs_adjacency_mod.F90
    map_viirs_cris_def.F90
    map_viirs_cris_mod.F90
    mapping_viirs_cris.fdf
src/preproc
  libatovpp
    cris1cld.F
```

4.2 Calling graph for main program



5 Results and Validation

The mapping has been tested over several Lannion local passes and a global pass which allows to test on some specific cases such as polar data. Run time is less than 30 second for a 15 minutes of M data, more time is spent in aggregation.

Validation is performed by comparing the VIIRS radiances spatially convolved in CrIS FOV and the CrIS 1c spectrum convolved with the VIIRS instrument spectral response functions.

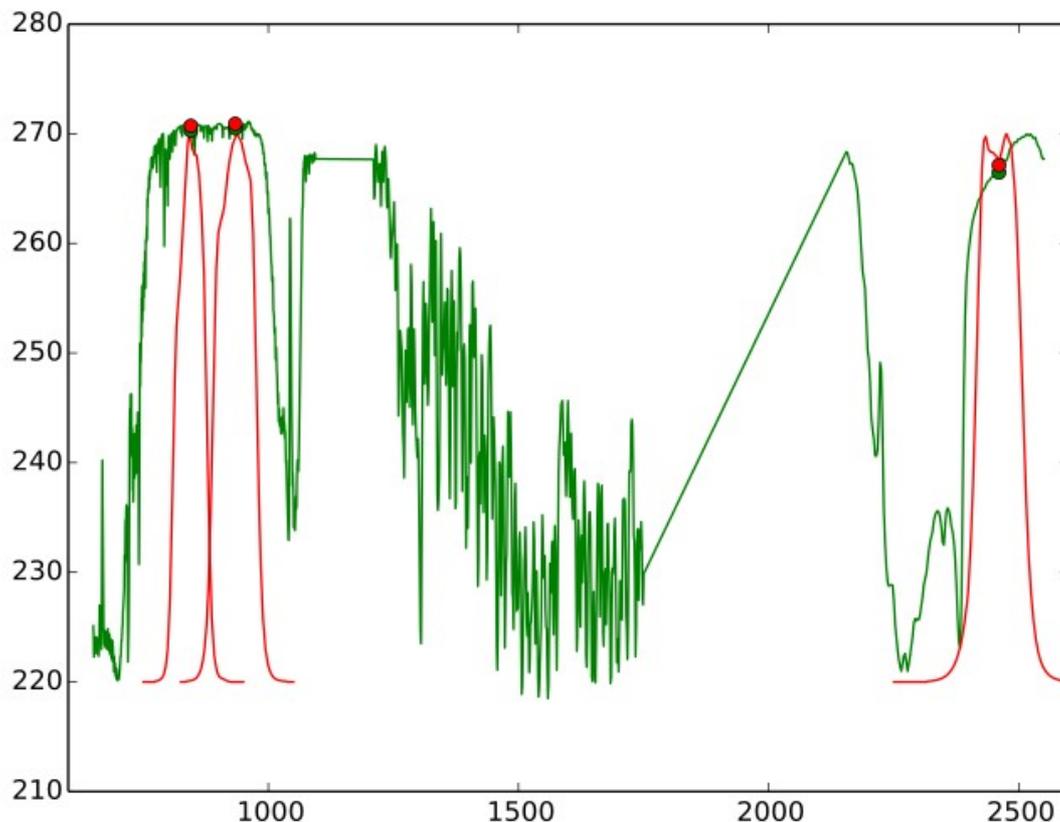
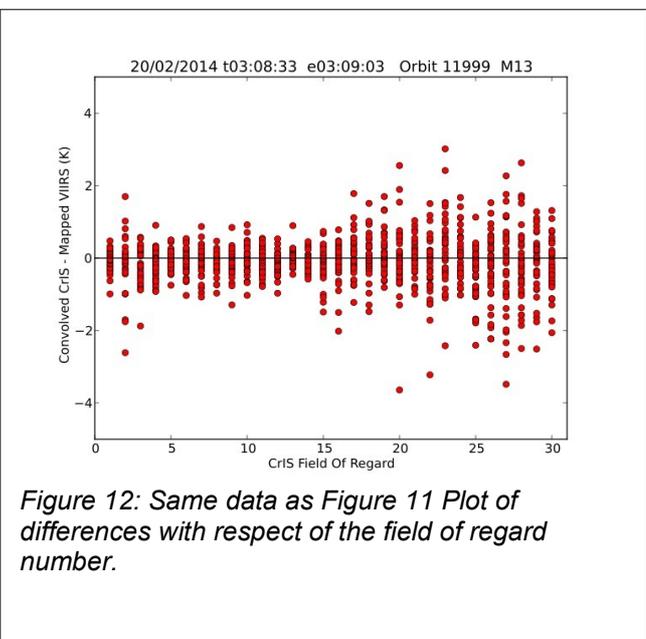
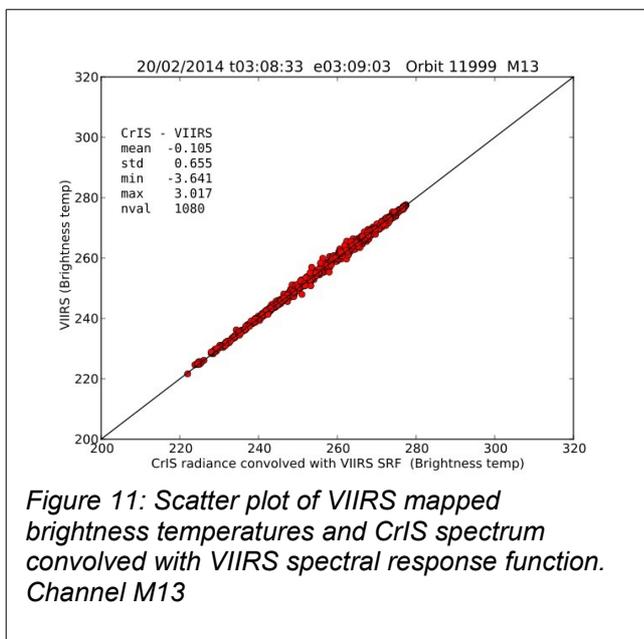


Figure 10: VIIRS SRFs (M13, M15 and M16) are shown above overlaid with a CrIS spectrum. Green circle: CrIS radiances convolved with VIIRS SRF. Red circle: VIIRS radiances mapped to CrIS FOV.

The VIIRS channels that lay in the CrIS spectrum are I5, M13, M15 and M16. All those channels are considered in the validation process, so band I and M are tested. Results show rms errors always below 0.8K. Any test trying to spatially shift the VIIRS data gives worse results. The small observed bias can be explained by the unit conversion of the VIIRS radiances, due to uncertainty of central frequencies used in ADL/CSPP for calibration and conversion algorithms.

It is shown that the VIIRS and CrIS geolocation are in good agreement and that the adjacency method is of great interest for mapping those two instruments.

The Figure 11 shows an example of scatter plots that we get for a CrIS granule when comparing the VIIRS results of the mapping algorithm and the CrIS 1c spectrum convolved with the VIIRS SRFs. The 4 infrared VIIRS channels show the same behaviour. For plotting one only retain CrIS FOVs that are entirely covered by VIIRS granules. The spread is due to cloudy scenes where the radiances inhomogeneity in the FOV and the FOV shape uncertainties do occur.



5.2 Test on full orbit

The orbit pass 15537 of the October 27th 2014 (11h33-13h25) has been used for testing (all VIIRS geolocation is terrain corrected). The first thing was to confirm the mapping is able to run over the poles and this is confirmed.

The main thing is that the results are almost same quality over the pass. But we can observe a general worse quality for field of regards that are in second part of the swath (field of regard 15 to 30). This needs more investigation, a comparison with locally received and processed data has been done. This test show the same behaviour. Most of the high difference values are observed for low brightness temperatures, but this is not the major effect. More study will be done, in particular focusing on central field of view.

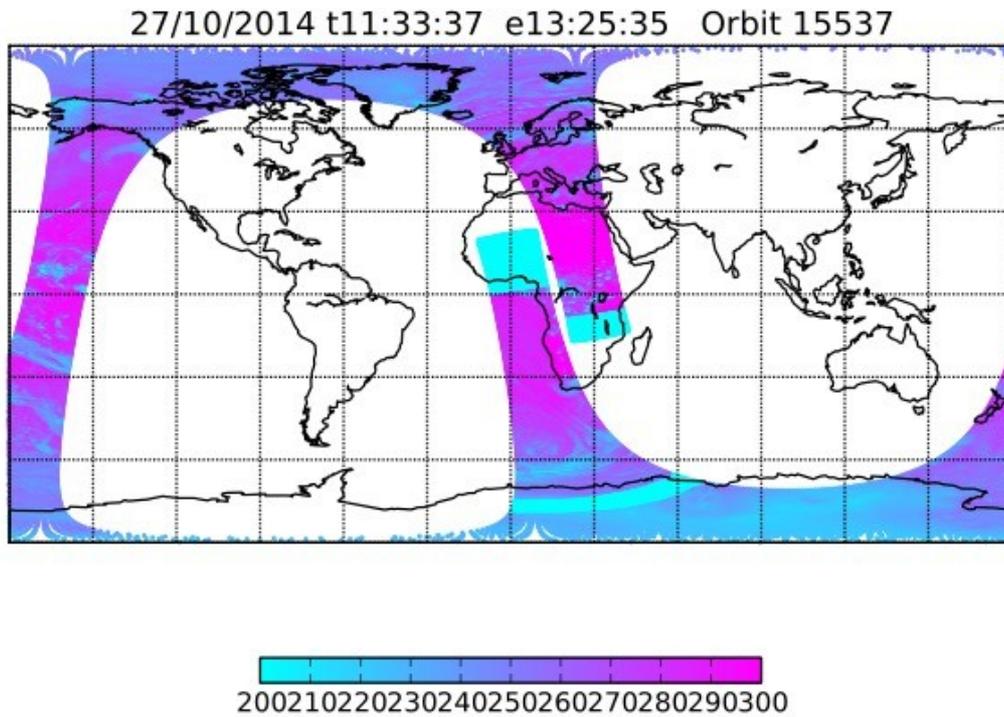


Figure 13: Full orbit pass 15537, VIIRS M16 brightness temperature mapped to CrIS. At beginning and end of orbit some areas without VIIRS data. Some missing VIIRS in ascending part, near South pole.

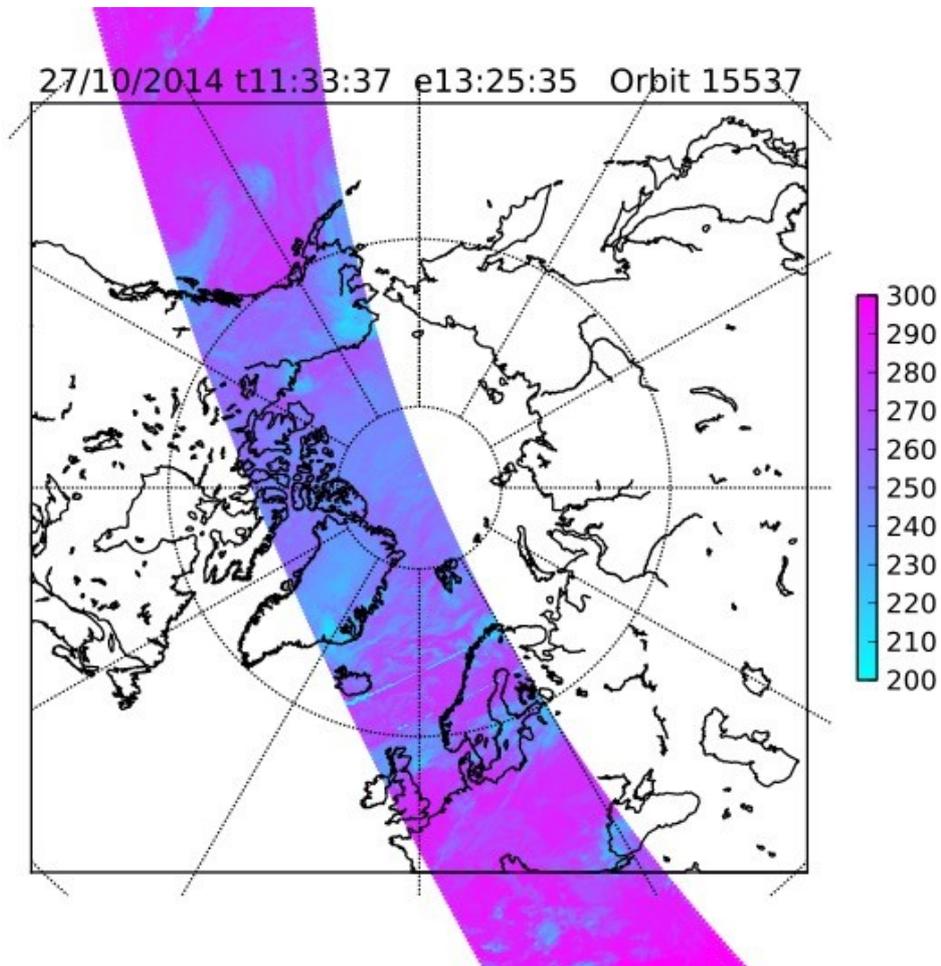


Figure 14: Zoom on North pole. Full orbit pass 15537, VIIRS M16 brightness temperature mapped to CrIS.

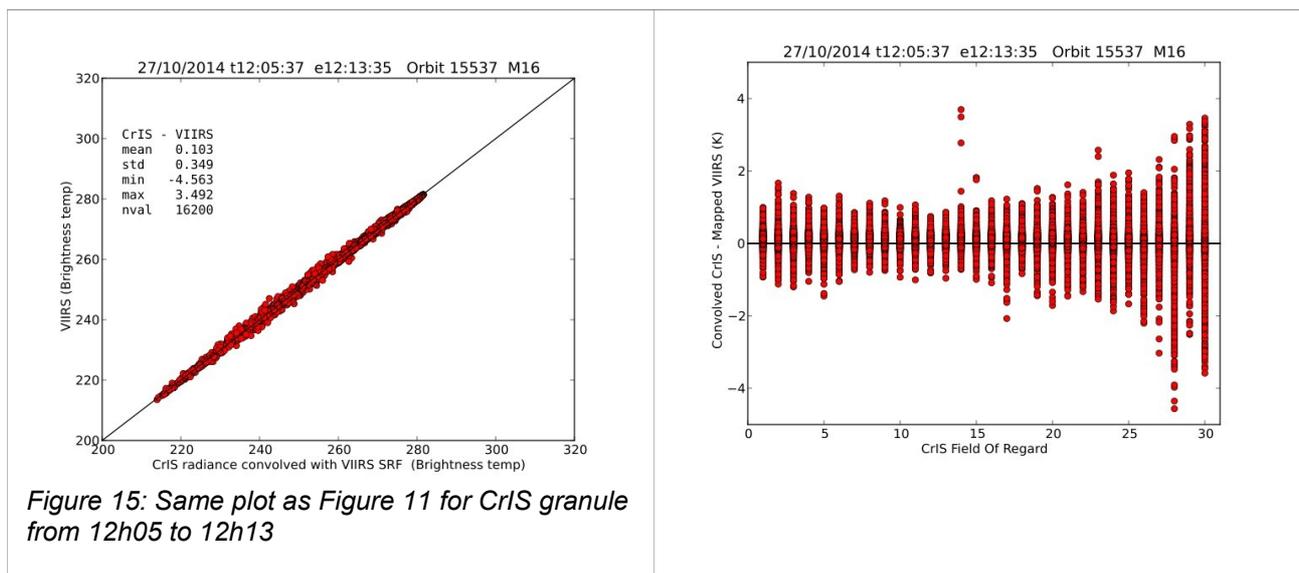
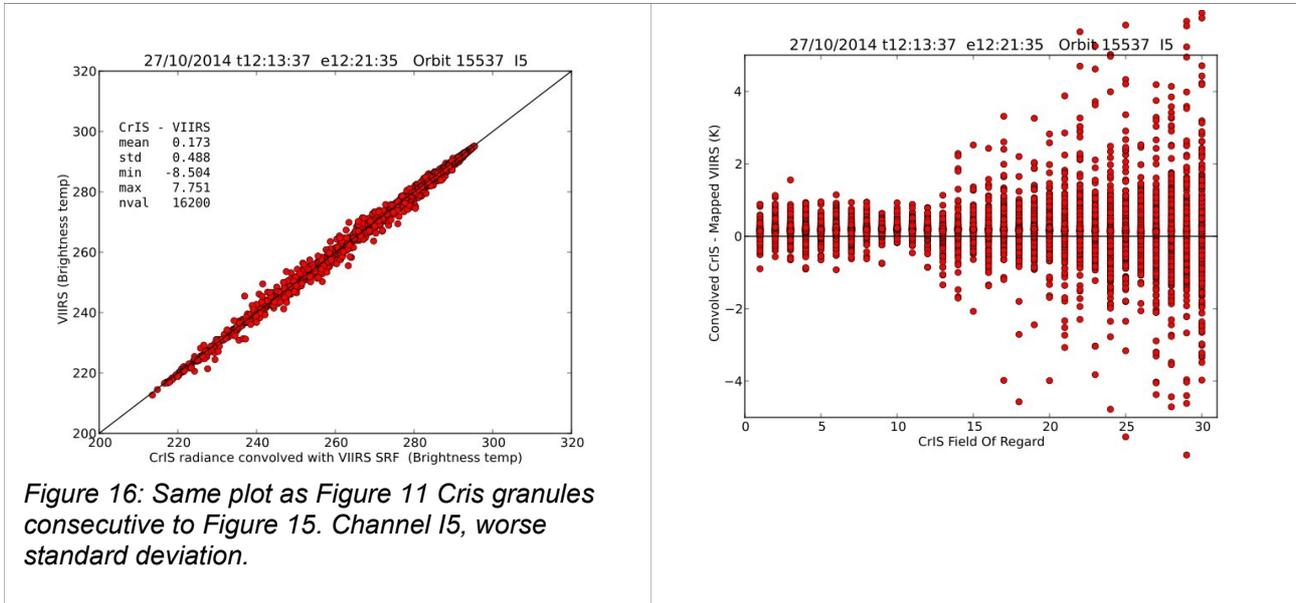


Figure 15: Same plot as Figure 11 for CrIS granule from 12h05 to 12h13



5.3 MAIA4 results

The MAIA4 cloud mask product file can be input to the program. Some of the information like cloud cover percentage, cloud top temperature and cloud top pressure can be stored in the level 1d CrIS file; see annex for full list of variables. The Figure 17 shows an example of MAIA4 cloud percentage mapped to the CrIS field of view.

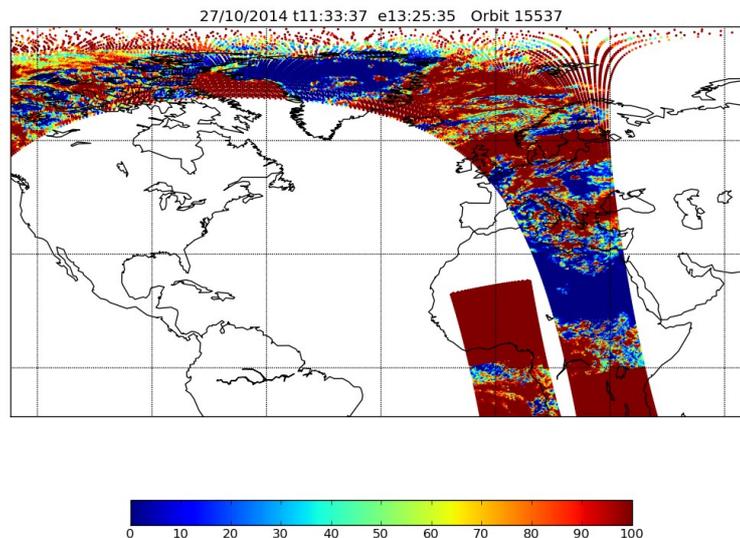


Figure 17: Maia4 cloud cover percentage for a part of the full orbit test case.

6 Conclusion and future work

We have shown that the VIIRS-CrIS mapping version 1.0 is able to provide results for all field of view and for any part of the world with quite good quality. The fortran source code has been highly tested over numerous CMS Lannion local passes and for a full orbit. Any improvement to the algorithm should not change the interfaces.

The worse quality for the field of regard 15 to 30 need more investigation, and this will be the subject for the next release.

The current release is good enough for users who wants to merge a VIIRS cloud mask such as MAAI4 with the CrIS radiances.

Some more work is also requested to improve the processing time spent by the geolocation terrain un-correction routine.

Annex AAPP I1c I1d variables updated by mapping

CrIS level 1C format

cris1c.h

integer*4 cris1c_landfrac(cris_1cnfov,cris_1cnfor) ! land fraction (0-100)
integer*4 cris1c_landsea(cris_1cnfov,cris_1cnfor) ! land/sea qualifier (0=land,
1=sea,2=coast)
integer*4 cris1c_cloudcover(cris_1cnfov,cris_1cnfor) ! cloud cover (percent)
integer*4 cris1c_cloudtopht(cris_1cnfov,cris_1cnfor) ! height of top of cloud (m) Not
available in MAIA4 output

CrIS level 1d format

cris1d.h

header record

! Channel order: 1-16 M band, 17-21 I band, 22 DNB

integer*4 cris1d_h_nviirschan !number of VIIRS channels in format
integer*4 cris1d_h_nviirschanused !number of channels used
integer*4 cris1d_h_viirschans(cris_1dnchanviirs) !Channel numbers
integer*4 cris1d_h_viirsbands(cris_1dnchanviirs) !Band numbers 5=M, 6=I, 7-DN

data record

integer*4 cris1d_cloudflag(10,cris_1dnfor)
o 1 % clear
o 2 % cloudy
o 3 % clear over snow
o 4 % clear over ice
o 5 % aerosol_dust_ash_fire
o 6 mean CLOUD Top Pressure (hPa * 10) Missing value : -9999
o 7 mean CLOUD Top Temperature (Celsius * 100) Missing value : -9999
o 8 mean surface temperature over sea (Celsius * 100) Missing value : -9999
o 9 mean surface temperature over land (Celsius * 100) Missing value : -9999
o 10 Pixel quality flags: % of good quality flags (fov_qual==0)
integer*4 cris1d_viirsradiance(cris_1dnchanviirs,cris_1dnfor)
integer*4 cris1d_viirsreflectance(cris_1dnchanviirs,cris_1dnfor)

units are mW/m².sr.cm⁻¹ for radiances