

**GSFC JPSS CMO  
November 15, 2023  
Released**

474-00448-03-06, Revision A  
Joint Polar Satellite System (JPSS) Code 474

**Joint Polar Satellite System  
Algorithm Specification Volume III:  
Operational Algorithm Description (OAD)  
for the VIIRS RDR/SDR**



NOAA / NASA

**Goddard Space Flight Center  
Greenbelt, Maryland**

**Joint Polar Satellite System  
Algorithm Specification Volume III:  
Operational Algorithm Description (OAD) for the VIIRS RDR/SDR  
Review/Signature/Approval Page**

**Prepared by:**

LEO Ground Services Project SE

**Approved by:**

Kellyann Jeletic  
LEO Ground Services Project SEIT Lead

Nicolaie Todirita  
LEO Ground Services Project Manager

**Electronic Approval available on-line at: [https://jpssmis.gsfc.nasa.gov/frontmenu\\_dsp.cfm](https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm)**

## Preface

This document is under JPSS Ground configuration control. Once this document is approved, JPSS approved changes are handled in accordance with Class I and Class II change control requirements as described in the JPSS Configuration Management Procedures, and changes to this document shall be made by complete revision.

Any questions should be addressed to:

JPSS Configuration Management Office  
NASA/GSFC  
Code 474  
Greenbelt, MD 20771

## Change History Log

<b>Revision</b>	<b>Effective Date</b>	<b>Description of Changes</b> (Reference the CCR & CCB/ERB Approval Date)
Rev -	Jul 26, 2013	This was approved by the JPSS Ground ERB via <b>474-CCR-13-1104</b> on the effective date shown.
Rev -1	Oct 23, 2014	This version incorporates <b>474-CCR-14-2094</b> which was approved by JPSS Ground ERB on the effective date shown. This version was baselined in support of Change Order 10.
Rev -1 - Revalidated	Dec 11, 2019	The expiration date on this document has been extended five years by 474-CCR-19-4789, which was approved by the JPSS Ground AERB on Dec 11, 2019. The effective date and the revision/version on this document will remain the same, only the expiration date has been changed.
Rev A	Aug 09, 2023	This version incorporates 474-CCR-23-6516 which was approved by the Ground ERB on May 01, 2023 and the Ground CCB on May 24, 2023 and 474-CCR-23-6461 which was approved by the Ground Segment ERB on Aug 04, 2023 and by the JPSS Ground Segment CCB on the effective date shown.

## Table of Contents

1	INTRODUCTION .....	1
1.1	Objective .....	1
1.2	Scope .....	1
2	RELATED DOCUMENTATION .....	2
2.1	Parent Documents .....	2
2.2	Applicable Documents .....	2
3	ALGORITHM OVERVIEW .....	3
3.1	VIIRS SDR Verified RDR Description .....	7
3.1.1	Interfaces .....	9
3.1.1.1	Inputs .....	9
3.1.1.2	Outputs .....	9
3.1.2	Algorithm Processing .....	9
3.1.2.1	createVerifiedRDR() .....	16
3.1.2.2	ProViirsVerifiedRDRConverter() .....	16
3.1.2.3	ViirsCalAppPacket() .....	16
3.1.2.4	ViirsEngAppPacket() .....	16
3.1.2.5	ViirsHrdAppPacket() .....	16
3.1.3	Graceful Degradation .....	16
3.1.3.1	Graceful Degradation Inputs .....	16
3.1.3.2	Graceful Degradation Processing .....	17
3.1.3.3	Graceful Degradation Outputs .....	17
3.1.4	Exception Handling .....	17
3.1.5	Data Quality Monitoring .....	17
3.1.6	Computational Precision Requirements .....	17
3.1.7	Algorithm Support Considerations .....	17
3.1.8	Assumptions and Limitations .....	17
3.2	VIIRS SDR Geolocation Description .....	17
3.2.1	Interfaces .....	18
3.2.1.1	Inputs .....	18
3.2.1.2	Outputs .....	19
3.2.1.2.1	Geolocation Gridded Outputs .....	20
3.2.2	Algorithm Processing .....	23
3.2.2.1	Main driver for Geolocation ( geolocateGranule() ) .....	23
3.2.2.2	GEO_absolute_limit_check() .....	25
3.2.2.3	GEO_determine_DNB_sample_time_offsets() .....	25
3.2.2.4	GEO_determine_sample_time_offsets() .....	26
3.2.2.5	GEO_determine_thermal_corrections() .....	26
3.2.2.6	GEO_determine_view_vectors() .....	26
3.2.2.7	GEO_evaluate_polynomial() .....	27
3.2.2.8	GEO_find_next_flag() .....	27
3.2.2.9	GEO_interp_mod_unagg() .....	27
3.2.2.10	GEO_interpolate_mirror_encoder() .....	28
3.2.2.11	GEO_interpolate_telescope_encoder() .....	28

3.2.2.12	GEO_process_parameters()	28
3.2.2.13	GEO_relative_limit_check()	28
3.2.2.14	GEO_validate_scan_encoder_data()	28
3.2.2.15	geolocatePixel()	28
3.2.2.16	geolocateDecim()	28
3.2.2.17	geolocateFullFromDecim()	28
3.2.2.18	calcModFromImg()	29
3.2.2.19	createInterpRctngls()	29
3.2.2.20	initGeoDataStructs()	29
3.2.2.21	storeGranule()	29
3.2.2.22	fixSatAngles()	29
3.2.2.23	quadInterp()	29
3.2.2.24	geolocateAllRecPix()	29
3.2.2.25	interpLocations()	29
3.2.2.26	interpAngles()	29
3.2.2.27	eclipseAndSaaFlags()	29
3.2.2.28	Geolocation Quality Flag Logic	30
3.2.2.29	Create Degree GEO Products	30
3.2.3	Graceful Degradation	30
3.2.3.1	Graceful Degradation Inputs	30
3.2.3.2	Graceful Degradation Processing	30
3.2.3.3	Graceful Degradation Outputs	30
3.2.4	Exception Handling	30
3.2.5	Data Quality Monitoring	31
3.2.6	Computational Precision Requirements	31
3.2.7	Algorithm Support Considerations	31
3.2.8	Assumptions and Limitations	31
3.3	VIIRS SDR Calibration Description	31
3.3.1	Interfaces	33
3.3.1.1	Inputs	33
3.3.1.2	Outputs	34
3.3.1.2.1	Definitions of the Calibration VIIRS SDR Outputs	36
3.3.1.2.2	VIIRS On-Board Calibrator Intermediate Product	36
3.3.2	Algorithm Processing	36
3.3.2.1	Sub-frame Offset Compensation	37
3.3.2.2	Robust Algorithm Design	38
3.3.2.3	Earth View Radiometric Calibration Error Handling	38
3.3.2.4	calibrateSDR()	38
3.3.2.5	Preprocess_RDR_Data() and Preprocess_DNB()	38
3.3.2.6	SDRSetup()	38
3.3.2.7	Combine_Reduced_Quality_Flags()	38
3.3.2.8	Calibrate_Emissive_Bands()	38
3.3.2.9	Find_SV_Scan_Index()	38
3.3.2.10	Calibrate_Reflective_Bands()	39
3.3.2.11	Calibrate_DayNight_Bands()	39
3.3.2.12	Aggregate_750M_DG()	39

3.3.2.13viirs_decmp()	39
3.3.3 Graceful Degradation	39
3.3.3.1 Graceful Degradation Inputs	39
3.3.3.2 Graceful Degradation Processing	39
3.3.3.3 Graceful Degradation Outputs	39
3.3.4 Exception Handling	39
3.3.5 Data Quality Monitoring	40
3.3.6 Computational Precision Requirements	40
3.3.7 Algorithm Support Considerations	40
3.3.8 Assumptions and Limitations	40
3.3.8.1 I/O Timeliness	40
3.3.8.2 Sci2Ops Issues	40
3.3.8.3 Additional Assumptions	40
3.3.8.4 Additional Limitations	41
3.4 VIIRS SDR Reflective Solar Band (RSB) Automated Calibration (RSBAutoCal)	
Description	41
3.4.1 Interfaces	41
3.4.1.1 Inputs	41
3.4.1.2 Outputs	41
3.4.2 Algorithm Processing	42
3.4.2.1 Automated Reflective Solar Band Calibration Data Processing Logic	42
3.4.2.2 Calibration Symbols and Units	43
3.4.2.3 Reflective Calibration Core Equations	45
3.4.2.4 Emissive Calibration Core Equations	45
3.4.2.5 Day Night Band Calibration Core Equations	46
3.4.2.6 Solar Diffuser View Data Processing Core Equations	46
3.4.2.7 Event Processing	47
3.4.2.7.1 Solar Event Processing Flow	47
3.4.2.7.1.1 calcOrbitalMedianAndSmoothH	48
3.4.2.7.1.2 calcOrbitalMedianAndSmoothRsbF	49
3.4.2.7.1.3 calcOrbitalMedianAndSmoothDnbLgsGain	49
3.4.2.7.1.4 calcOrbitalMedianAndSmoothDnbDarkSignal	49
3.4.2.7.1.5 calcOrbitalMedianAndSmoothDnbGainRatios)	49
3.4.3 Graceful Degradation	50
3.4.3.1 Graceful Degradation Inputs	50
3.4.3.2 Graceful Degradation Processing	50
3.4.3.3 Graceful Degradation Outputs	50
3.4.4 Exception Handling	50
3.4.5 Data Quality Monitoring	50
3.4.6 Computational Precision Requirements	50
3.4.7 Algorithm Support Considerations	50
3.4.8 Assumptions and Limitations	50
3.5 VIIRS SDR Bright Pixel Description	50
3.5.1 Interfaces	51
3.5.1.1 Inputs	51
3.5.1.2 Outputs	51

3.5.2	Algorithm Processing .....	52
3.5.2.1	Estimating % Scattered Light .....	52
3.5.2.2	“Calculation Unreliable” Flag .....	54
3.5.2.3	Creation of Non-Saturated Scene .....	54
3.5.2.4	Common Adjacency Use In Bright Pixel .....	55
3.5.2.5	Data Convolution .....	56
3.5.3	Graceful Degradation .....	56
3.5.3.1	Graceful Degradation Inputs.....	56
3.5.3.2	Graceful Degradation Processing .....	56
3.5.3.3	Graceful Degradation Outputs .....	56
3.5.4	Exception Handling .....	56
3.5.5	Data Quality Monitoring .....	57
3.5.6	Computational Precision Requirements .....	57
3.5.7	Algorithm Support Considerations.....	57
3.5.8	Assumptions and Limitations .....	57
4	GLOSSARY/ACRONYM LIST .....	58
4.1	Glossary .....	58
4.2	Acronyms .....	61

## List of Figures

Figure 3-1.	IPO Model Interface to INF and DMS (Calibration) .....	3
Figure 3-2.	VIIRS Verified RDR Flow Diagram.....	4
Figure 3-3.	VIIRS Geolocation Flow Diagram.....	5
Figure 3-4.	VIIRS Calibration Flow Diagram .....	6
Figure 3-5.	VIIRS RSBAutoCal Flow Diagram .....	6
Figure 3-6.	VIIRS Bright Pixel Flow Diagram.....	7
Figure 3.1-1.	Instrument Engineering Order and Product Order Illustration .....	8
Figure 3.1.2-1.	Create Verified VIIRS RDR Flow Diagram .....	10
Figure 3.2-1.	Geolocation Processing Chain .....	18
Figure 3.2.2.1-1.	geolocateGranule() Flow Diagram.....	25
Figure 3.2.2.6-1.	GEO_determine_view_vectors() Block Diagram (Geolocation).....	27
Figure 3.3-1.	Calibration Processing Chain.....	33
Figure 3.3.2-1.	Radiometric Calibration Main Program Flow.....	37
Figure 3.4.2.1-1.	High level RSBAutoCal Data Processing Main Program Flow.....	42
Figure 3.4.2.7.1-1.	Detailed Solar Event Processing Program Flow .....	48

## List of Tables

Table 3.1.1.1-1.	VIIRS Verified RDR Inputs .....	9
Table 3.1.2-1.	Verified VIIRS RDR Extracted Engineering Packet Data .....	11
Table 3.1.2-2.	Verified VIIRS RDR Extracted Science Packet Data .....	15
Table 3.1.2-3.	Verified VIIRS RDR Extracted Calibration Packet Data.....	15
Table 3.2.1.1-1.	VIIRS SDR Geo Inputs .....	18
Table 3.2.1.2-1.	VIIRS Non Gridded SDR Geo Outputs.....	19



---

Table 3.2.1.2.1-1. Map Data Set (mds_type) (Geolocation).....	20
Table 3.2.1.2.1-2. Gridded Geolocation Moderate Output (Geolocation).....	22
Table 3.2.1.2.1-3. Gridded Geolocation Imagery Output (Geolocation).....	22
Table 3.2.1.2.1-4. Gridded Geolocation Day Night Band Output (Geolocation).....	22
Table 3.3.1.1-1. Earth View Radiometric Calibration Unit Inputs.....	34
Table 3.3.1.2-1. Earth View Radiometric Calibration Unit Outputs .....	35
Table 3.4.1.1-1. VIIRS SDR RSBAutoCal Inputs.....	41
Table 3.4.1.2-1. VIIRS SDR RSBAutoCal Outputs .....	41
Table 3.4.2.2-1. Calibration Symbols and Units.....	43
Table 3.4.2.3-1. Reflective Calibration Core Equations (Calibration) .....	45
Table 3.4.2.4-1. Emissive Calibration Core Equations (Calibration).....	45
Table 3.4.2.5-1. Day Night Band Calibration Core Equations (Calibration) .....	46
Table 3.4.2.6-1. Solar Diffuser View Data Processing Core Equations (Calibration) .....	46
Table 3.5.1.1-1. VIIRS SDR Bright Pixel Inputs .....	51
Table 3.5.1.2-1. VIIRS Bright Pixel Output .....	52



# 1 INTRODUCTION

## 1.1 Objective

The purpose of this Operational Algorithm Description (OAD) document is to express, in computer-science terms, the remote sensing algorithms that produce the Joint Polar Satellite System (JPSS) end-user data products. These products are individually known as Raw Data Records (RDRs), Temperature Data Records (TDRs), Sensor Data Records (SDRs) and Environmental Data Records (EDRs). In addition, any Intermediate Products (IPs) produced in the process. The OAD provides a software description of that science as implemented in the operational ground system.

The purpose of an OAD is two-fold:

1. Provide initial implementation design guidance to the operational software developer.
2. Capture the “as-built” operational implementation of the algorithm reflecting any changes needed to meet operational performance/design requirements.

An individual OAD document describes one or more algorithms used in the production of one or more data products. This document is limited to the description of the core operational algorithm(s) required to create the GEO, CAL, RSBAutoCal, OBC-IP and Bright Pixel products for the VIIRS SDR. It provides a general overview and is intended to supplement in-line software documentation and interface control documentation for maintenance of the operational software.

## 1.2 Scope

The purpose of the Operational Algorithm Description (OAD) document is to the VIIRS-Bright\_pixel IP, the VIIRS-RSBAutocal-History-IP, the VIIRS SDRs, and the VIIRS GEO.

## 2 RELATED DOCUMENTATION

The latest JPSS document(s) can be obtained from URL:

[https://jpssmis.gsfc.nasa.gov/frontmenu\\_dsp.cfm](https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm). JPSS Project documents have a document number starting with 470, 472 or 474 indicating the governing Configuration Control Board (CCB) (Program, Flight, or Ground) that has the control authority of the document.

### 2.1 Parent Documents

The following reference document is the Parent Document from which this document has been derived. Any modification to a Parent Document will be reviewed to identify the impact upon this document. In the event of a conflict between a Parent Document and the content of this document, the JPSS Program CCB has the final authority for conflict resolution.

Document Number	Title
474-00448-01-06	JPSS Algorithm Specification Volume I: SRS for the VIIRS RDR/SDR

### 2.2 Applicable Documents

The following documents are the Applicable Documents from which this document has been derived. Any modification to an Applicable Document will be reviewed to identify the impact upon this document. In the event of conflict between an Applicable Document and the content of this document, the JPSS Program Control Board has the final authority for conflict resolution.

Document Number	Title
429-05-02-42	NPP Mission Data Format Control Book (MDFCB)
429-05-02-42-02	NPP MDFCB Appendix A
472-00251	Mission Data Format Control Book (MDFCB) Joint Polar Satellite System-1 (JPSS-1)
474-00448-02-01	JPSS Algorithm Specification Volume II: Data Dictionary for the Common Algorithms
474-00448-02-06	JPSS Algorithm Specification Volume II: Data Dictionary for the VIIRS RDR/SDR
474-00448-02-08	JPSS Algorithm Specification Volume II: Data Dictionary for the Common Geolocation and Spacecraft Orientation

### 3 ALGORITHM OVERVIEW

To begin data processing, the VIIRS SDR algorithms (Verified RDR, Geolocation and Calibration) are initiated or retasked by the Infrastructure (INF) Subsystem Software Item (SI). The INF SI provides tasking information to the VIIRS SDR algorithms indicating which granule number and version to process. The Data Management Subsystem (DMS) SI provides data storage and retrieval capability. A library of C++ classes is used to implement the SI interfaces (for more information refer to the JPSS CGS IDPS PRO Software User Manual Part 2, document UG60917-IDP-1005, particularly sections regarding PRO Common processing and the IPO Model). The VIIRS SDR algorithms are identified as a persistent retaskable process. When it completes processing for a specified granule, instead of shutting down, the algorithm requests new tasking data in order to process another granule. Figure 3-1 shows the IPO (Input, Processing, Output) model interface to INF and DMS for calibration only but the Verified RDR and Geolocation algorithms follow the same IPO model.

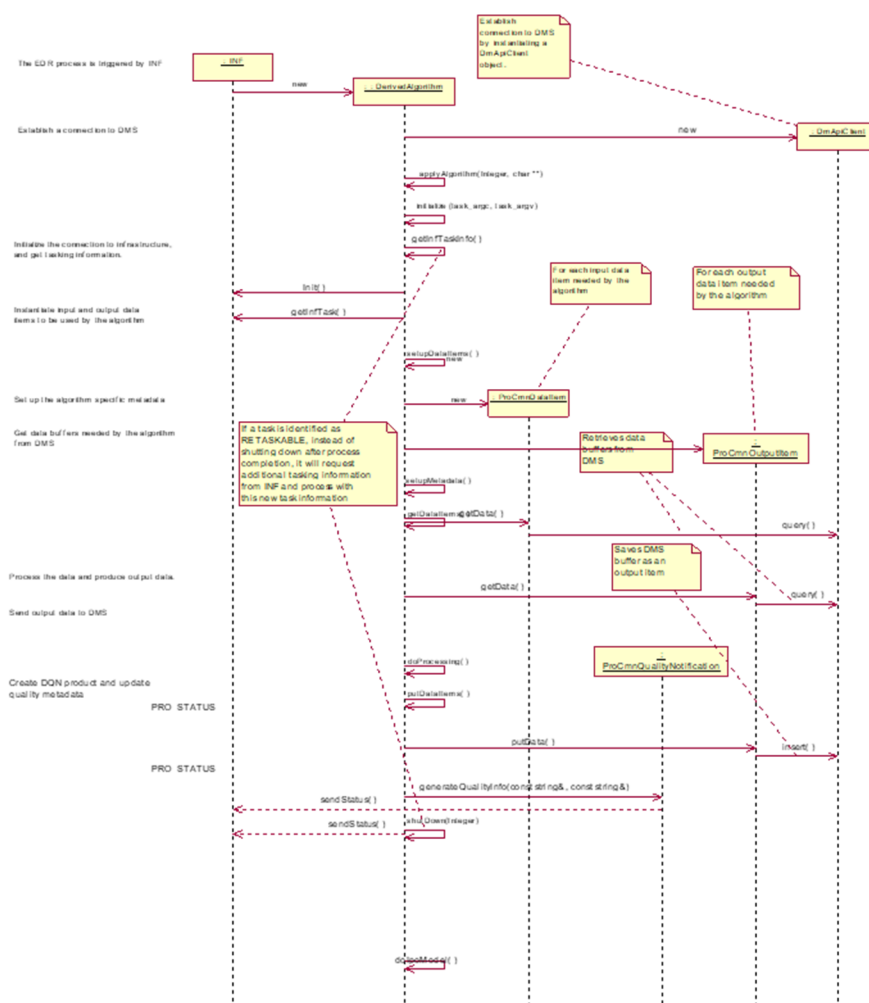


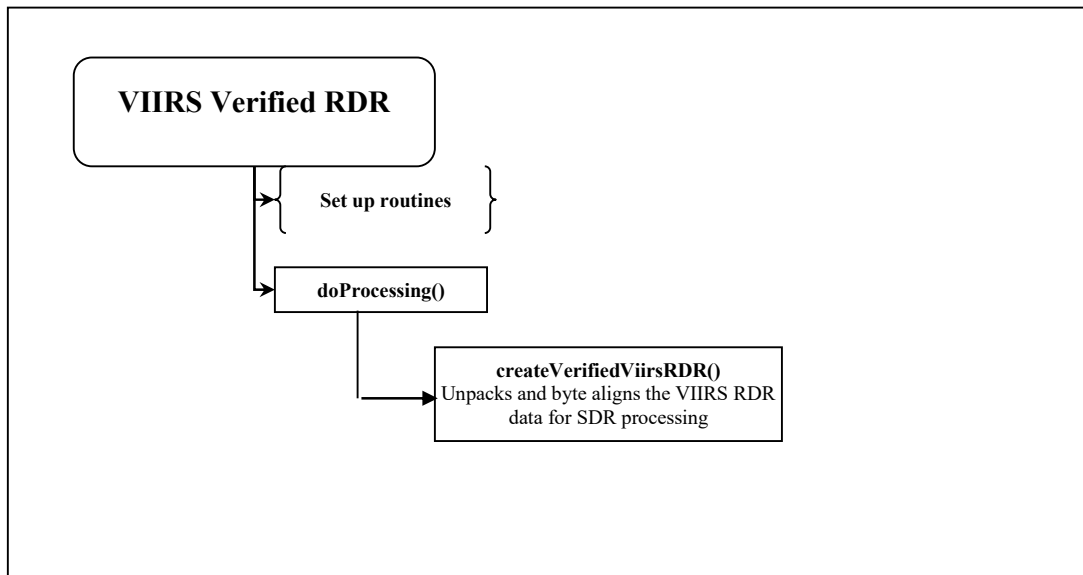
Figure 3-1. IPO Model Interface to INF and DMS (Calibration)

When the VIIRS SDR algorithms are called, they first initialize values such as routine name and software version number, then call `setupDataItems()` to create data items handlers for retrieving input data from, and storing output data to, DMS. Next, `doPtrAssignmentToInputAndOutput()` assigns pointers to input and output data items. The Verified RDR algorithm, `ProSdrViirsVerifiedRDR`, produces the Verified RDR. The geolocation algorithm, `ProSdrViirsGeo` controls the execution and processing of the geolocation outputs.

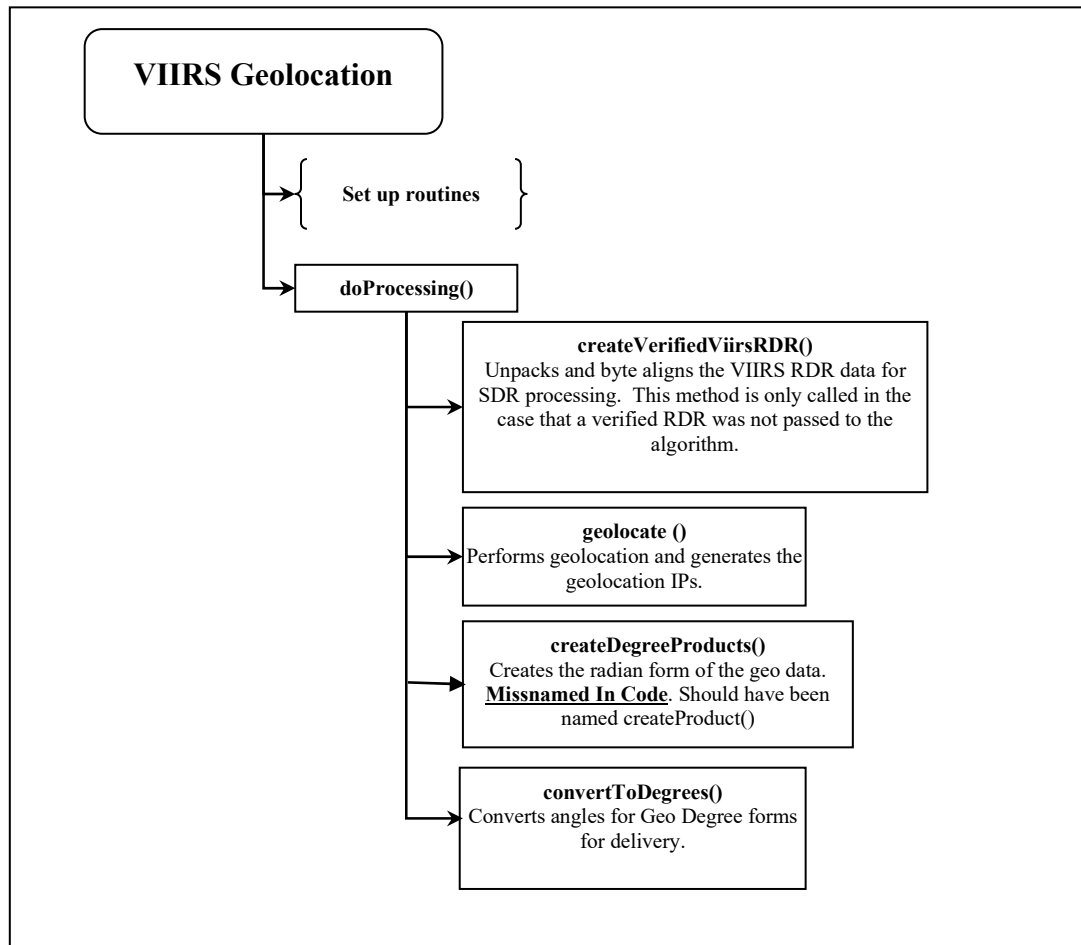
In the `ProSdrViirsGeo` algorithm, consecutive calls are made to the `GEO` method, `geolocateGranule`, with appropriate resolution parameter structures to produce GEO products in the following order (first to last): imagery (non-terrain corrected and terrain corrected items); moderate (non-terrain corrected and terrain corrected items); moderate unaggregated; DNB (non-terrain corrected and terrain corrected items).

The `ProSdrViirsCal` algorithm calls the Radiometric Calibration routine to produce TOA radiances, reflectances (reflective bands only), and brightness temperatures (emissive bands only) for each pixel in the imagery bands, moderate bands, and day/night band.

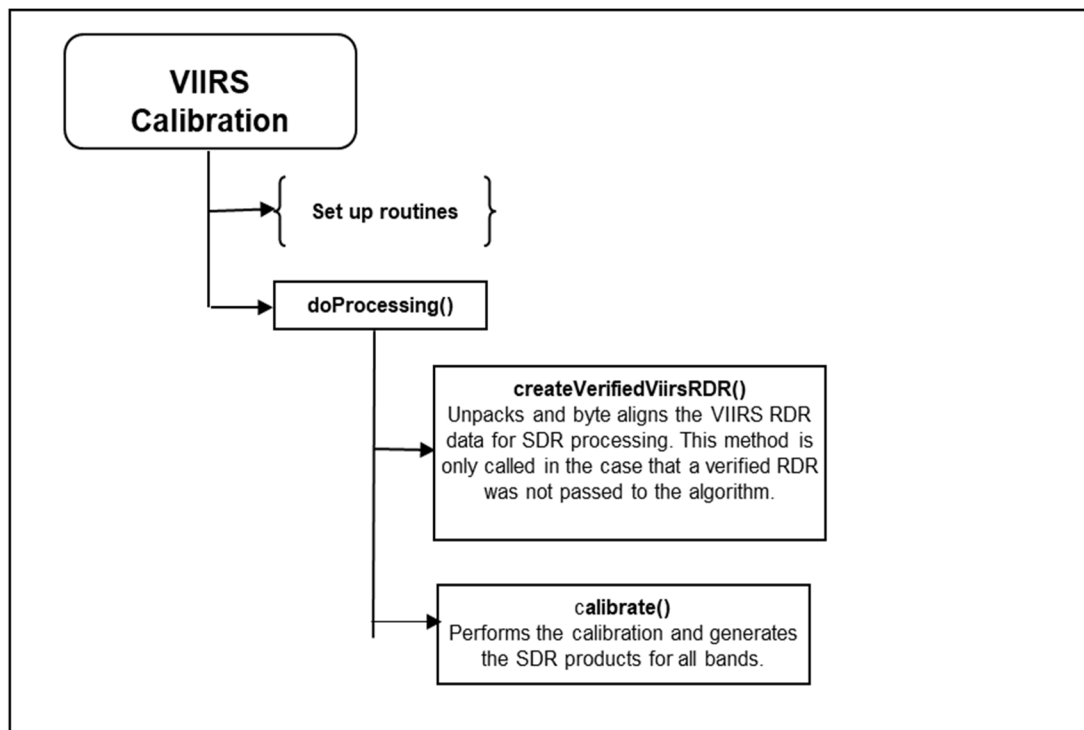
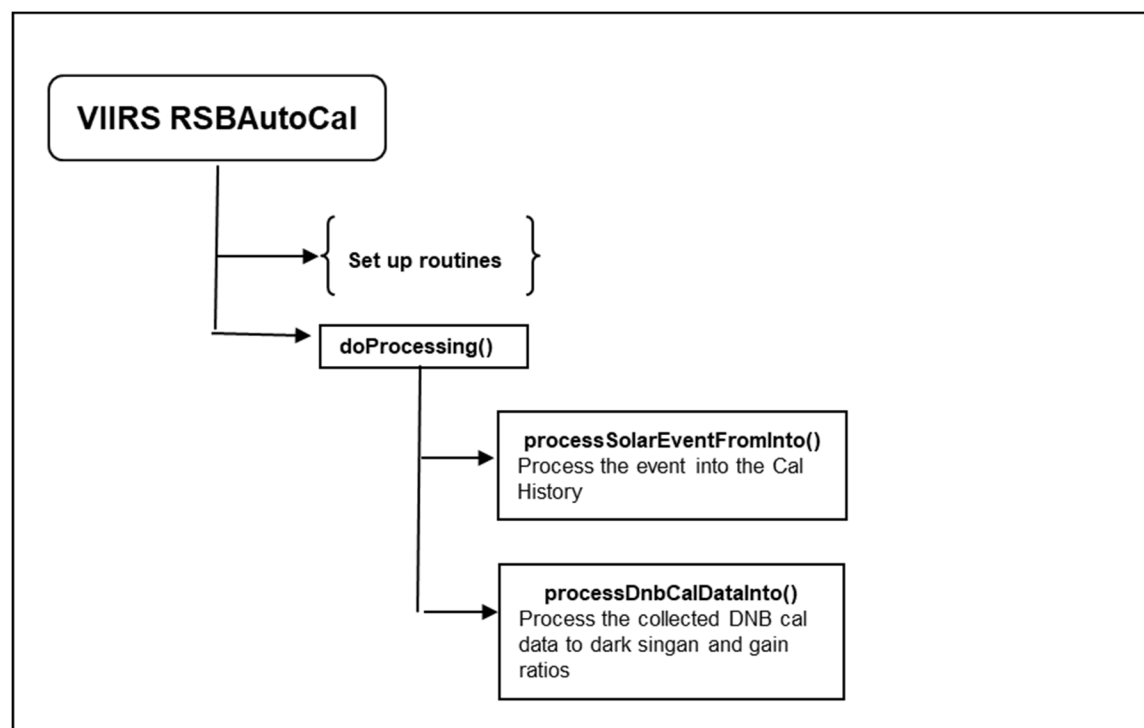
Finally, outputs are checked for data quality, and select outputs are scaled before all output data items are written to DMS. A flow diagram of the VIIRS SDR algorithms is provided in Figure 3-2 and Figure 3-3 through Figure 3-6 for clarification.



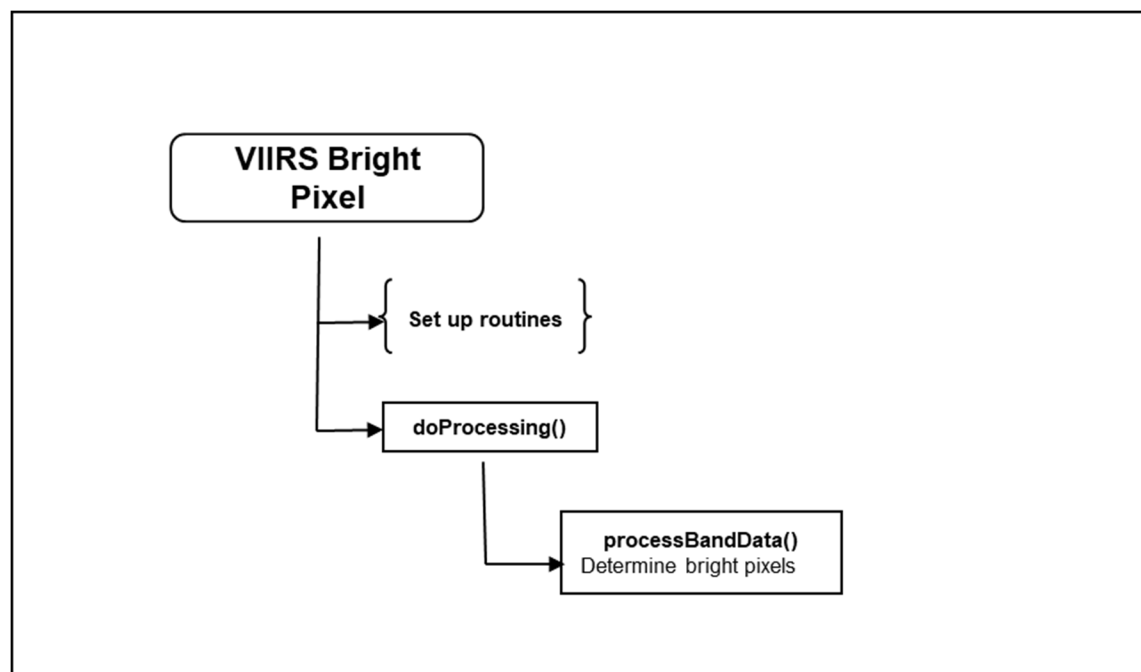
**Figure 3-2. VIIRS Verified RDR Flow Diagram**



**Figure 3-3. VIIRS Geolocation Flow Diagram**

**Figure 3-4. VIIRS Calibration Flow Diagram****Figure 3-5. VIIRS RSBAutoCal Flow Diagram**





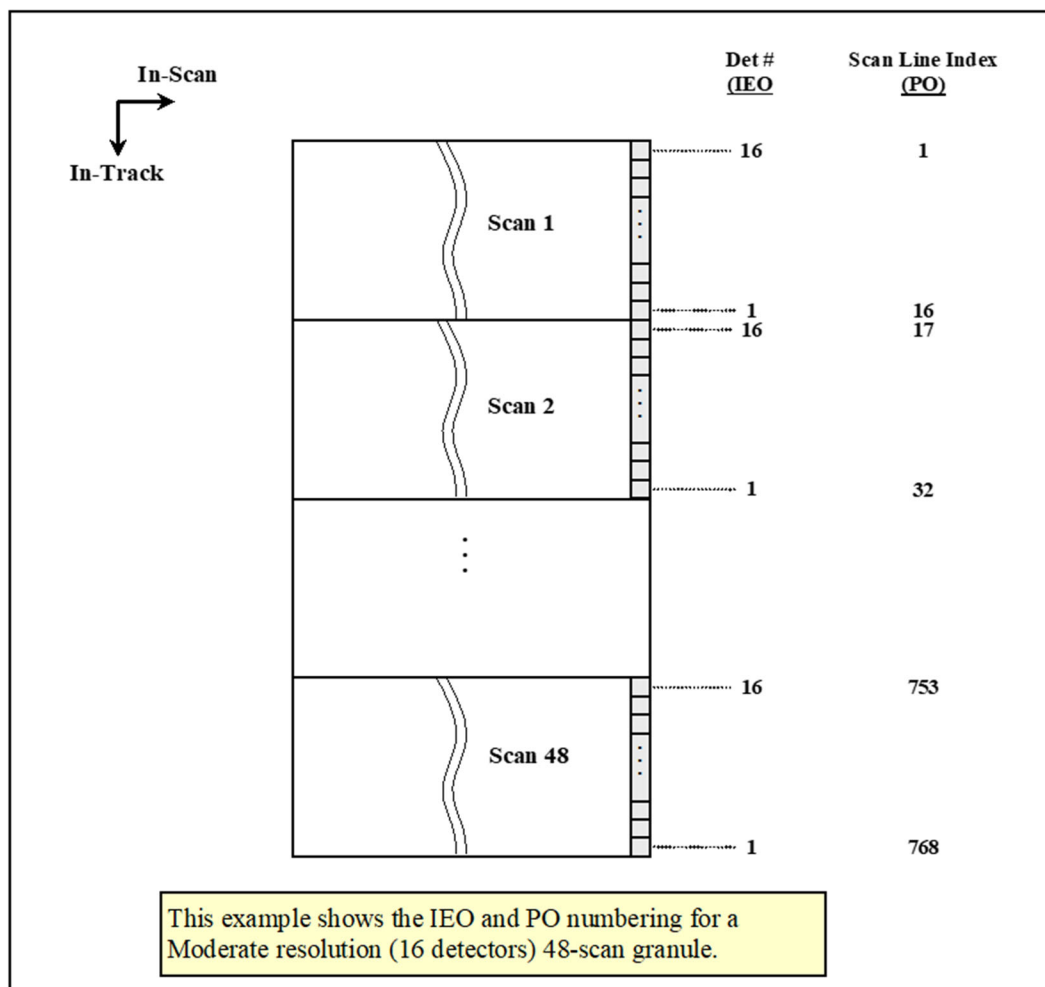
**Figure 3-6. VIIRS Bright Pixel Flow Diagram**

### 3.1 VIIRS SDR Verified RDR Description

Before the geolocation or calibration components can be run the RDR packets must be assembled into what is called the Verified RDR. The RDR packets must first be decompressed. There are four stages of decompression. All stages are lossless. The stages in order of application on the compression side for the earth-view (EV) data are:

- Removal of unused bits
- Piecewise linear offset for dual-gain bands only
- Differential Encoding using a predictor band
- Universal Source Encoder for Space (USES), an industry standard for lossless encoding.

The decompression reverses the compression steps and is therefore applied in the reverse of the order listed above. The decompressed EV packets are then assembled into matrices in the order needed by the calibration algorithm. The calibration views must also be assembled into matrices called granules. The granules have  $n$  full scans (either  $n_{scan}=47$  or  $48$  in the current configuration). The scans are assembled from first to last in time, with increasing scan line indices corresponding to later times. This is referred to as the in-track index. The detector focal plane arrays (FPA) are positioned so that detector 1 is at the leading edge of the FPA in the track direction. Unfortunately, this is the opposite of what is needed to produce the verified RDR with the scan lines increasing in the in-track direction. Therefore, the order of the detectors must be reversed in creating granules from packets. The numbering of the detectors on the FPA is referred to as Instrument Engineering Order (IEO). The reversed ordering that is used in the verified RDR (as well as the SDR and EDR) is referred to as Product Order (PO). All look-up tables that have per detector values should be in Product Order. Figure 3-7 provides an illustration of the different detector ordering.



**Figure 3.1-1. Instrument Engineering Order and Product Order Illustration**

The other dimension is in the scan direction, referred to as the scan index, and has increasing indices from start of scan to end of scan. The imagery bands (I1 to I5) have 6400 samples in the scan index for the EV verified RDR. The dual gain moderate resolution bands (M1 to M5, M7 and M13) have 6304 samples and the other moderate resolution bands have 3200. The DNB has 4064 EV samples in-scan.

The Cal views have 96 samples in scan for imagery bands I1 to I5, and 48 samples in-scan for the moderate resolution bands. The number of samples per view for the DNB band is 64 except for Aggregation Modes 34 and 35 when it is 16. The DNB aggregation mode in the Cal view changes every two scans, cycling over aggregation Modes 1 to 36 in 72 scans. The DNB Cal data is split between the four FPAs used by the DNB: HGA, HGB, MGS, and LGS. Because the M16 detectors use TDI, it contains calibration data for both M16A and M16B detectors. The EV data, however, is aggregated before transmission for M16, so there is no separate A and B data in the verified RDR. When the Verified RDR granule arrays are created, fill values must be put in pixels where there is missing data.

Verified engineering application packet data is also produced and byte aligned as part of this process.

For the geolocation algorithm the Verified Spacecraft E&A RDR is generated from the RDR packets. This is S/C Ephemeris and Attitude RDR data that has been byte aligned. This process is done as part of the geolocation algorithm. The onboard S/C ephemeris and attitude data (in APID 11) includes: spacecraft ECR (GPS ECEF) position vectors (X, Y, Z components in meters), ECR velocity vectors (X, Y, Z components in meters per second) and attitude quaternion data (4 dimensionless numbers). The attitude quaternion data are used to create a quaternion matrix providing unambiguous rotation from J2000 ECI coordinates to spacecraft coordinates.

### 3.1.1 Interfaces

#### 3.1.1.1 Inputs

VIIRS Science RDRs contain the science, CAL and engineering application packets that are input to the VIIRS Verified RDR, in Table 3.1.1.1-1, and the VIIRS SDRs. The flow is in Table 3-1 of 474-00448-01-06.

**Table 3.1.1.1-1. VIIRS Verified RDR Inputs**

Input	Description	Reference Document
VIIRS RDR	VIIRS Science RDR	474-00448-02-06_JPSS-DD-Vol-II-Part-06

#### 3.1.1.2 Outputs

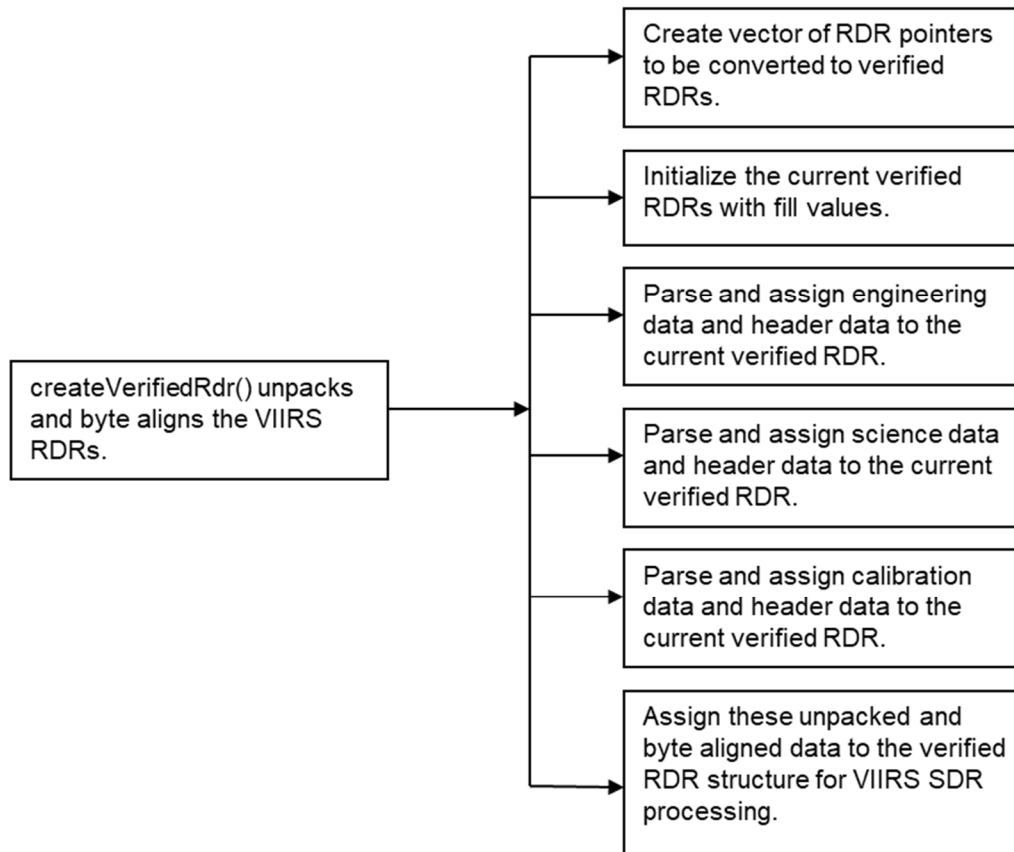
The internal output (heap item) of the VIIRS Verified RDR is a byte-aligned version of the VIIRS RDR and is not detailed in the SRS or DD.

### 3.1.2 Algorithm Processing

This is the derived algorithm for the VIIRS SDR Verified RDR algorithm and is a subclass of the AutoGeneratedProSdrViirsVerifiedRdr and ProCmnAlgorithm classes. The derived algorithm class creates a list of input data items read from DMS and passes required data into the algorithm. All output data items are written to DMS once the algorithm finishes processing this data.

VIIRS Science RDRs contain the science, CAL and engineering application packets that are required for SDR processing of a tasked granule. In addition to the tasked granule, the RDR information from the previous and following granule also need to be processed in order to accurately calibrate dual gain bands. Not all of the data in the RDRs are needed for SDR processing. A Verified RDR contains only the data from an RDR that is required for the SDR processing of a tasked granule. The data has been extracted, unpacked and byte aligned from the CCSDS application packets contained in the RDR. In other words, a Verified RDR is a byte-aligned RDR. This unpacked and byte-aligned data is assigned to the internal verified RDR structure, which is not in the same structure that the packet data came in (for efficiency reasons), and VIIRS SDR processing uses the data from that structure to do further processing. Figure 3.1.2-1 shows the Create Verified VIIRS RDR Flow.

The first sample from each of the 16 DNB detector outputs reported in the Earth View and the three calibration views (Space, Blackbody and Solar Diffuser) is anomalous for each VIIRS scan. The main cause for this anomaly was found to be due to a problem in the startup portion of the CCD timing. This timing problem may be fixed in the hardware for Flight Unit 2; however, for Flight Unit 1, this DNB anomaly should be removed during the VIIRS Verified RDR process by placing fill integer 0 values in the RDR data for the DNB Earth View and the three calibration views.



**Figure 3.1.2-1. Create Verified VIIRS RDR Flow Diagram**

The contents of the unpacked RDR data packets are listed below in Table 3.1.2-1, 3.1.2-2, and 3.1.2-3.

**Table 3.1.2-1. Verified VIIRS RDR Extracted Engineering Packet Data**

Field	Type/ Dimensions	Description	Units/ Valid Range	Fill Value
startIETusec	Int64/ (nscans)	IET time at the start of the scan which is the number of microseconds since 1/1/1958	IET microseconds	MISS_IN T64
startTAIsec	Float64/ (nscans)	TAI time at the start of the scan which is the number of seconds since 1/1/1958	TAI seconds	MISS_FL OAT64
mirror_side	uint8/ (nscans)	mirror side in optical path – “0” for side 1 and “1” for side 2	Unitless/ [0 1]	MISS_UI NT8
dpp_config	uint8/ (nscans)(128)	digital PreProcessor Configuration	Unitless/ Count	MISS_UI NT8
mf_ao_blkhd_px_nz	int16/ (nscans)	MF_AO_BLKHD_PX_NZ Therm 44	Unitless/ Count	MISS_IN T16
mf_ao_blkhd_nx_pz	int16/ (nscans)	MF_AO_BLKHD_NX_PZ Therm 45	Unitless/ Count	MISS_IN T16
mf_stopassy_baff_nz	int16/ (nscans)	On Baffle NZ of Apert Stop Assy between HAM and FM2 Therm 14	Unitless/ Count	MISS_IN T16
mf_fold_mir_bkhd_ct	int16/ (nscans)	MF_FOLD MIRROR BLKHD Center Therm 6	Unitless/ Count	MISS_IN T16
mf_tel_blkhd_py	int16/ (nscans)	Mainframe Telescope Bulkhead Therm 33, RT16 (Node 1022 in Radiometry Model)	Unitless/ Count	MISS_IN T16
ap_lw_cca	int16/ (nscans)	Long wave IR CCA Information stored in the Engineering packet.	Unitless/ Count	MISS_IN T16
ap_sm_cca	int16/ (nscans)	Short/Med wave IR CCA information stored in the Engineering packet.	Unitless/ Count	MISS_IN T16
ap_vn_cca	int16/ (nscans)	Visible/Near IR CCA information stored in the Engineering packet	Unitless/ Count	MISS_IN T16
ct_prec_tref_mux1ca1	int16/ (nscans)	Precision Thermistor Cal Ref Resistor 1	Unitless/ Count	MISS_IN T16
ct_prec_tref_mux1ca2	int16/ (nscans)	Precision Thermistor Cal Ref Resistor 2	Unitless/ Count	MISS_IN T16
ct_prec_tref_mux1ca3	int16/ (nscans)	Precision Thermistor Cal Ref Resistor 3	Unitless/ Count	MISS_IN T16
ft_lw_cfpa_hi_rsl	int16/ (nscans)	focal plane Temperature Controller Long Wave IR CFPA high resolution temperature	degrees Kelvin/ 70 – 89K Note 1	MISS_IN T16
ft_lw_cfpa_lo_rsl	int16/ (nscans)	focal plane Temperature Controller Long Wave IR CFPA wide range temperature	degrees Kelvin/ 52 – 330K Note 1	MISS_IN T16
ft_sm_cfpa_hi_rsl	int16/ (nscans)	focal plane Temperature Controller Short Midwave IR CFPA high resolution temperature	degrees Kelvin/ 70 – 89K Note 1	MISS_IN T16
ft_sm_cfpa_lo_rsl	int16/ (nscans)	focal plane Temperature Controller Short Midwave IR CFPA wide range temperature	degrees Kelvin/ 51- 330K Note 1	MISS_IN T16

Field	Type/ Dimensions	Description	Units/ Valid Range	Fill Value
ft_vis_nir_fpa	int16/ (nscans)	focal plane Temperature Controller Visual/Near IR fpa	degrees Kelvin/ 245- 319K Note 1	MISS_IN T16
m1_asp_offset	int16/ (nscans)(2)( 16)	moderate band 1 Analog Signal Processor offset	Unitless	MISS_IN T16
m2_asp_offset	int16/ (nscans)(2)( 16)	moderate band 2 Analog Signal Processor offset	Unitless	MISS_IN T16
m3_asp_offset	int16/ (nscans)(2)( 16)	moderate band 3 Analog Signal Processor offset	Unitless	MISS_IN T16
m4_asp_offset	int16/ (nscans)(2)( 16)	moderate band 4 Analog Signal Processor offset	Unitless	MISS_IN T16
m5_asp_offset	int16/ (nscans)(2)( 16)	moderate band 5 Analog Signal Processor offset	Unitless	MISS_IN T16
m6_asp_offset	int16/ (nscans)(2)( 16)	moderate band 6 Analog Signal Processor offset	Unitless	MISS_IN T16
m7_asp_offset	int16/ (nscans)(2)( 16)	moderate band 7 Analog Signal Processor offset	Unitless	MISS_IN T16
m8_asp_offset	int16/ (nscans)(2)( 16)	moderate band 8 Analog Signal Processor offset	Unitless	MISS_IN T16
m9_asp_offset	int16/ (nscans)(2)( 16)	moderate band 9 Analog Signal Processor offset	Unitless	MISS_IN T16
m10_asp_offset	int16/ (nscans)(2)( 16)	moderate band 10 Analog Signal Processor offset	Unitless	MISS_IN T16
m11_asp_offset	int16/ (nscans)(2)( 16)	moderate band 11 Analog Signal Processor offset	Unitless	MISS_IN T16
m12_asp_offset	int16/ (nscans)(2)( 16)	moderate band 12 Analog Signal Processor offset	Unitless	MISS_IN T16
m13_asp_offset	int16/ (nscans)(2)( 16)	moderate band 13 Analog Signal Processor offset	Unitless	MISS_IN T16
m14_asp_offset	int16/ (nscans)(2)( 16)	moderate band 14 Analog Signal Processor offset	Unitless	MISS_IN T16
m15_asp_offset	int16/ (nscans)(2)( 16)	moderate band 15 Analog Signal Processor offset	Unitless	MISS_IN T16
m16a_asp_offset	int16/ (nscans)(2)( 16)	moderate band 16 Analog Signal Processor offset	Unitless	MISS_IN T16
m16b_asp_offset	int16/ (nscans)(2)( 16)	day/night band Analog Signal Processor offset	Unitless	MISS_IN T16
i1_asp_offset	int16/ (nscans)(2)( 32)	imagery band 1 Analog Signal Processor offset	Unitless	MISS_IN T16
i2_asp_offset	int16/ (nscans)(2)( 32)	imagery band 2 Analog Signal Processor offset	Unitless	MISS_IN T16
i3_asp_offset	int16/ (nscans)(2)( 32)	imagery band 3 Analog Signal Processor offset	Unitless	MISS_IN T16
i4_asp_offset	int16/ (nscans)(2)( 32)	imagery band 4 Analog Signal Processor offset	Unitless	MISS_IN T16
i5_asp_offset	int16/ (nscans)(2)( 32)	imagery band 5 Analog Signal Processor offset	Unitless	MISS_IN T16
bb_tmps	int16/ (nscans)(6)	black body temperatures, OT shutdown=315K	degrees Kelvin/ 243 to 336 K Note 1	MISS_IN T16
ham_tmp1	int16/ (nscans)	half angle mirror T1 prec therm 07, radiatively coupled	degrees Celsius/ -30 to 65 °C Note 1	MISS_IN T16

Field	Type/ Dimensions	Description	Units/ Valid Range	Fill Value
ham_tmp2	int16/ (nscans)	half angle mirror T2 prec therm 08, radiatively coupled	degrees Celsius/ -30 to 65 °C Note 1	MISS_IN T16
mf_nadir_rad_nxp	int16/ (nscans)	mainframe Nadir radiator nxpy	degrees Celsius/ -25 to 65 °C Note 1	MISS_IN T16
mf_scan_cavity_nxp	int16/ (nscans)	mainframe scan cavity nx pz therm 46	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
mf_scan_cavity_baf_nz	int16/ (nscans)	mainframe scan cavity baffle NZ prec therm 10	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
mf_scan_cavity_baf_pz	int16/ (nscans)	mainframe scan cavity baffle PZ prec therm 9 temperature	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
mf_scan_cavity_bknd_n	int16/ (nscans)	mainframe scan cavity bulkhead ny therm 8	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
dp_dnb_cca	int16/ (nscans)	digital preprocessor DNB cca thermistor 60	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
dp_dpp_cca	int16/ (nscans)	digital preprocessor DPP cca thermistor 61	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
dp_fpie_clk_cca	int16/ (nscans)	digital preprocessor focal plane interface electronics clk cca thermistor 51	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
power_supply1	int16/ (nscans)	power supply 1 thermistor 52	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
power_supply2	int16/ (nscans)	power supply 2 thermistor 53	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
se_a_cca	int16/ (nscans)	scan control electronics A cca thermistor 31	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
se_b_cca	int16/ (nscans)	scan control electronics B cca thermistor 12	degrees Celsius/ -25 to 60°C Note 1	MISS_IN T16
sdsd_position	UInt8/ (nscans)	SDSM position 0=home, 1=SD view and 2=sun view	Unitless/ 0- 2	MISS_UI NT8

Field	Type/ Dimensions	Description	Units/ Valid Range	Fill Value
sdsd_samples	int16/ [nscans] [n_sdsd_samples(5)] [n_sdsd_detectors(8)]	SDSM Samples	Volts/ -2.5 to 2.5 V Note 1	MISS_IN T16
sdsd_preamp	int16/ [nscans]	SDSM Preamp Thermistor 57	degrees Celsius/ -70 to 120°C Note 1	MISS_IN T16
tel_enc	uint16/ [nscans x (num. of TEL enc. Pulses)]	contains the raw encoder values for each scan in the granule	Scan time counter ticks/[0 – 65535]	MISS_IN T16
ham_enc	uint16/ [nscans x (num. Of HAM enc. Pulses)]	contains the raw encoder values for each scan in the granule	Scan time counter ticks/[0 – 65535]	MISS_IN T16
tel_start_enc	uint16 / [nscans]	Telescope start of scan encoder value (15-bit encoder)	15-bit encoder ticks/ [0 – 32768]	MISS_IN T16
ham_start_enc	uint16 / [nscans]	Half angle mirror start of scan encoder value (15-bit encoder)	15-bit encoder ticks/ [0 – 32768]	MISS_IN T16
act_scans	int32/ 1	number of actual scans	N/A/ 0 – 16	MISS_IN T32
scan_mode	char8/ [nscans]	sensor mode of each scan	N/A/ [Day   Night   Other]	MISS_UI NT8
Mode	uint8/ 1	scan mode	NA/ 0=night, 1=day, 2=mixed	MISS_UI NT8
bgnTime	int64/ 1	begin boundary of the granule	IET	MISS_IN T64
endTime	int64/ 1	end boundary of the granule	IET	MISS_IN T64
first_scan	int64/ 1	IET of the first scan	IET	MISS_IN T64
last_scan	int64/ 1	IET of the last scan	IET	MISS_IN T64
sensor_softver	uint8/ [16]	sensor software version	N/A	MISS_UI NT8
Sensor	uint8/ [8]	VIIRS sensor ID	N/A	MISS_UI NT8
sensor_model	int32/ 1	1=EDU, 2=FU1 (Flight Unit), FU2, FU3	N/A	MISS_IN T32
dnb_sequence	uint8/ [nscans]	DNB sequence number	Unitless/ 1 to 36	MISS_UI NT8

Note: The field is a digital count. Units/Valid Range are after polynomial conversion.



**Table 3.1.2-2. Verified VIIRS RDR Extracted Science Packet Data**

Field	Type Dimensions	Description	Units Range	Fill Value
<b>Image Band</b>				
IETusec	Int64 [48]	IET start of the scan	Microseconds since Jan 1, 1958	MISS_INT64_FILL
earthview	UInt16 / [1536][6400]	each pixel of the scan	12 bit count 0 – 4095	MISS_UINT16_FILL
<b>Single Gain Moderate Band</b>				
IETusec	Int64 [48]	IET start of the scan	Microseconds since Jan 1, 1958	MISS_INT64_FILL
earthview	UINT16 [768][3200]	each pixel of the scan	12 bit count 0 – 4095	MISS_UINT16_FILL
<b>Dual Gain</b>				
IETusec	Int64 [48]	IET start of the scan	Microseconds since Jan 1, 1958	MISS_INT64_FILL
earthview	UInt16 [768][6304]	each pixel of the scan	12 bit count 0 – 4095	MISS_UINT16_FILL
gain	UInt8 [768][6304]	the gain for each pixel	0 or 1	MISS_UINT8_FILL
<b>DNB</b>				
IETusec	Int64 [48]	IET start of the scan	Microseconds since Jan 1, 1958	MISS_INT64_FILL
earthview	UInt16 [768][4064]	each pixel of the scan	13 or 14 bit count 0 – 16383	MISS_UINT16_FILL
Gain	UInt8 [768][4064]	the gain for each pixel	0, 2, or 3	MISS_UINT8_FILL

**Table 3.1.2-3. Verified VIIRS RDR Extracted Calibration Packet Data**

Field	Type Dimensions	Description	Units Range	Fill Value
<b>Image Band</b>				
sv_calib	UInt16 [1536][96]	space view calibration source	counts 0 – 32767	MISS_UINT16_FILL
bb_calib	UInt16 [1536][96]	black body calibration source	counts 0 – 32767	MISS_UINT16_FILL
sd_calib	UInt16 [1536][96]	solar diffuser calibration source	counts 0 – 32767	MISS_UINT16_FILL
<b>Single Gain Moderate Band</b>				
sv_calib	UInt16 [768][48]	space view calibration source	counts 0 – 32767	MISS_UINT16_FILL
bb_calib	UInt16 [768][48]	black body calibration source	counts 0 – 32767	MISS_UINT16_FILL
sd_calib	UInt16 [768][48]	solar diffuser calibration source	counts 0 – 32767	MISS_UINT16_FILL
<b>Dual Gain / DN Bands</b>				
sv_calib	UInt16 [768][48]	space view calibration source	counts 0 – 32767	MISS_UINT16_FILL
bb_calib	UInt16 [768][48]	black body calibration source	counts 0 – 32767	MISS_UINT16_FILL
sv_calib	UInt16 [768][48]	solar diffuser calibration source	counts 0 – 32767	MISS_UINT16_FILL
sv_calib_gain	UInt8 [768][48]	gain state for the space view calibration	counts 0 – 1	MISS_UINT8_FILL

Field	Type Dimensions	Description	Units Range	Fill Value
bb_calib_gain	UInt8 [768][48]	gain state for the black body calibration	counts 0 – 1	MISS_UINT8 _FILL
sd_calib_gain	UInt8/ [768][48]	gain state for the solar diffuser calibration	counts 0 – 1	MISS_UINT8 _FILL
<b>DNB</b>				
sv_calib	UInt16 [768][64]	space view calibration source	counts 0 – 32767	MISS_UINT1 6_FILL
bb_calib	UInt16 [768][64]	black body calibration source	counts 0 – 32767	MISS_UINT1 6_FILL
sv_calib	UInt16 [768][64]	solar diffuser calibration source	counts 0 – 32767	MISS_UINT1 6_FILL

### 3.1.2.1 createVerifiedRDR()

createVerifiedRDR() creates a VIIRS Verified RDR structure for the current, previous, and following granules and calls ProViirsVerifiedRDRConverter::convert() to fill the Verified RDRs with unpacked and byte-aligned engineering, science, and CAL data. The Verified RDR number of actual scans is validated and the granule mode (day, night, mixed) is determined.

### 3.1.2.2 ProViirsVerifiedRDRConverter()

ProViirsVerifiedRDRConverter() retrieves the engineering, science, and CAL packets by APID from the VIIRS RDRs and calls the appropriate methods (ViirsEngAppPacket::parsePkts(), ViirsSciAppPacket::parsePkts(), ViirsEngCalPacket::parsePkts()) to unpack and byte align the data which is stored in the VIIRS Verified RDR. To ensure that every granule starts at a scan 1 and that missing scans are treated appropriately, scan ids are turned into offsets from the minimum scan id in the granule.

### 3.1.2.3 ViirsCalAppPacket()

The ViirsCalAppPacket class contains methods to retrieve and decompress the CAL packet data in the VIIRS RDR. It loops through the space view, black body, and solar diffuser CAL data and decompresses the views for each detector of each band. Universal Source Encoder for Space (USES) (viirs\_decmp()) is used for this purpose. This is also referred to Rice compression (after its creator or Modified NCSA (National Center for Supercomputing Applications) code. This class also converts the CCSDS time to IET time.\

### 3.1.2.4 ViirsEngAppPacket()

The ViirsEngAppPacket class contains methods to unpack and byte align engineering packet data in the VIIRS RDR. This class also retrieves and converts the CCSDS time to IET time.

### 3.1.2.5 ViirsHrdAppPacket()

The ViirsHrdAppPacket class contains methods to retrieve and decompress the science packet data in the VIIRS RDR. This class also converts the CCSDS time to IET time and fills in the start of scan trigger time for the RDR.

## 3.1.3 Graceful Degradation

The VIIRS Verified RDR procedure contains no graceful degradation.

### 3.1.3.1 Graceful Degradation Inputs

None.

### 3.1.3.2 Graceful Degradation Processing

None.

### 3.1.3.3 Graceful Degradation Outputs

None.

### 3.1.4 Exception Handling

No exception handling is performed in the VIIRS Verified RDR process.

### 3.1.5 Data Quality Monitoring

No data quality monitoring is performed in the VIIRS Verified RDR process.

### 3.1.6 Computational Precision Requirements

There are no computational precision requirements for the VIIRS Verified RDR process.

### 3.1.7 Algorithm Support Considerations

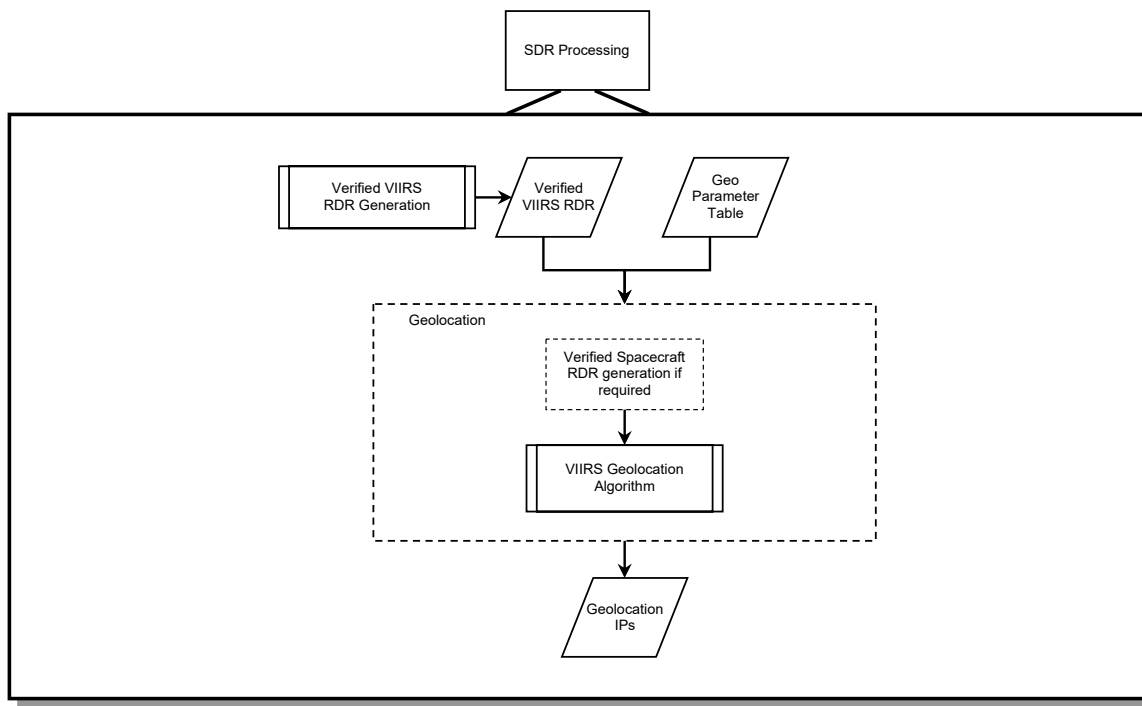
The DMS and INF must be running before the algorithm is executed.

### 3.1.8 Assumptions and Limitations

The VIIRS Verified RDR requires as input raw data packets transmitted from the satellite.

## 3.2 VIIRS SDR Geolocation Description

The purpose of the VIIRS GEO software is to provide Earth location and related spatial information for each of the samples in the Raw Data Record (RDR) belonging to the moderate resolution radiometric bands (aggregated and unaggregated), the fine resolution imagery bands, and the Day/Night Band (DNB). The output GEO data fields include geodetic latitude, longitude and derived products such as satellite zenith and azimuth angles, range to the satellite, solar zenith and azimuth angles, and (for the DNB) both lunar zenith and azimuth angles. The VIIRS GEO algorithm processes the engineering VIIRS RDR and Spacecraft Ephemeris and Attitude (E&A) RDRs, as well as external support data sets to generate the GEO data fields. The Earth location fields, which include the geodetic latitude, longitude and height, are computed using rigorous models of the Earth and VIIRS instrument pointing. The algorithm computes the earth location by first determining each pixel's line-of-sight (LOS) vector by constructing dynamic models of the VIIRS instrument using the engineering and spacecraft RDR data. Each detector's LOS vector is then intersected with the rotating WGS84 ellipsoid to obtain geodetic latitude and longitude. If terrain correction is to be performed, then the LOS intersection with the terrain-adjusted surface (geoid plus height over the geoid) is computed to output a terrain corrected geodetic latitude, longitude and height. The GEO products are held in internal memory to be input to the radiometric CAL module before they are written out to the Data Management Subsystem (DMS). For a list of GEO products to be produced, please refer to Section 3.2.1.2-1. The Geolocation Processing Chain is shown in Figure 3.2-1.

**Figure 3.2-1. Geolocation Processing Chain**

### 3.2.1 Interfaces

#### 3.2.1.1 Inputs

474-00448-01-06\_JPSS-SRS-Vol-I-Part-06, Table 3-1 (rows 4 - 7) lists the VIIRS GEO SDR inputs, which are further detailed within 474-00448-02-06\_JPSS-DD-Vol-II-Part-06, Section 4.3.2. The rotating telescope assembly (RTA) encoder data and half angle mirror (HAM) encoder data is used by the Geolocation algorithm to construct models of the RTA and HAM rotation as a function of time. The scan timing information is used to relate the VIIRS spatial elements to the models of both the dynamic components of the instrument (RTA and HAM) and the S/C. Some parameters listed in this table are not used by GEO but must remain in the data structure for sizing purposes. 474-00448-02-08\_JPSS-DD-Vol-II-Part-08, Section 4.2.2 contains a list of the S/C ephemeris and attitude data used to construct a model of the platform position, velocity, and orientation as a function of time. 474-00448-02-06\_JPSS-DD-Vol-II-Part-06, Section 7.1.27, Geolocation Parameter Inputs, contains instrument constants that include focal plane, band, and detector locations which are used to generate corrections applicable to each detector (channel) in a spatial element. Inputs are listed below:

**Table 3.2.1.1-1. VIIRS SDR Geo Inputs**

Input	Description	Reference Document
Verified VIIRS RDR	VIIRS Engineering, Science and Calibration Packet Data	See Section 3.1.2 of this document.
Spacecraft E&A RDR	Spacecraft Attitude and Ephemeris RDR	474-00448-02-08_JPSS-DD-Vol-II-Part-08
VIIRS GEO Parameter LUT	Contains the coefficients used in the geolocation algorithm.	474-00448-02-06_JPSS-DD-Vol-II-Part-06

## 3.2.1.2 Outputs

The VIIRS SDR GEO code produces seven non-gridded GEO products (plus QFs):

**Table 3.2.1.2-1. VIIRS Non Gridded SDR Geo Outputs**

Output	Description	Reference Document
VIIRS Mod Geo	VIIRS M-Band SDR Geolocation	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Img Geo	VIIRS I-Band SDR Geolocation	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS DNB Geo	VIIRS Day/Night Band SDR Geolocation	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Mod Unagg	VIIRS SDR Un-aggregated 750m Dual-Gain Band GEO	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Mod Terrain Corrected	VIIRS M-Band SDR Geolocation	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Img Terrain Corrected	VIIRS I-Band SDR Geolocation	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS DNB Terrain Corrected	VIIRS Day/Night Band SDR Geolocation	474-00448-02-06_JPSS-DD-Vol-II-Part-06

Of the seven non-gridded GEO products, the data is output in both radians and degrees. Either the radians geo product or the degrees geo product can be used by downstream algorithms. The two gridded GEO outputs are used for internal processing only. The default grid projection used by the Operational code is polar stereographic projection. The gridded GEO outputs are used for the remapping of Imagery EDRs and can be used for remapping from the VIIRS SDR to any map projection. See Section 3.2.1.2.1 for associated geolocation output data.

Each non-gridded product contains the granule scan and pixel level data. Note that for MOD, IMG, and DNB, the same structures are used for both terrain corrected and non-terrain corrected products. In the terrain corrected products, the latitude and longitude, and all of the pixel level products are adjusted for terrain.

Granule-Level quality flags (QF) (Automatic Quality and the Percent Missing) (note that these are two of the name/value pair metadata QFs) apply to IMG, MOD, and DNB bands and are defined below:

Automatic Quality QF (see above for references):

- 1) HAM/RTA Encoder Flag: If the encoder flags are set to bad or degraded, Automatic QF should be triggered.
- 2) HAM/RTA Encoder Flag: If the encoder data is missing, Automatic QF should be triggered.
- 3) Missing E&A Flag: If E&A data is missing, Automatic QF should be triggered.
- 4) SAA flag - Automatic QF should NOT be triggered as this does not affect the geo quality,
- 5) Eclipse flag - Automatic QF should NOT be triggered as this does not affect the geo quality.

- 6) Bad pointing QF and Percent out of Bounds - if this is triggered Automatic QF should be triggered.
- 7) Bad terrain and invalid solar angles QF - if these are triggered; the Automatic QF should be triggered.

Percent Missing QF (See 474-00448-02-06\_JPSS-DD-Vol-II-Part-06, Section (MOD) 6.2.68; (IMG) Section 6.3.24 (DNB) Section 6.4.8).

This quality flag is a percentage summary of:

- 1) geo Scan Quality=missing, OR
- 2) scan-level Missing ephemeris or attitude QF is anything other than nominal (there are 4 options)

#### 3.2.1.2.1 Geolocation Gridded Outputs

Each gridded product contains map data defined in Table 3.2.1.2.1-1 plus the pixel level data listed in Tables 3.2.1.2.1-2, 3.2.1.2.1-3, and 3.2.1.2.1-4.

**Table 3.2.1.2.1-1. Map Data Set (mds\_type) (Geolocation)**

Output	Type/Dimensions	Description	Units/Valid Range	Fill Value
grid_type	Int16	identifies which map projection is being used	11=std Mercator, cylindrical, coaxial 21=Polar Stereographic, Northern Hemisphere 25=Polar Stereographic, Southern Hemisphere 31=Lambert Conformal Conic, Tangent cone, Northern Hemisphere 35=Lambert Conformal Conic, Tangent cone, Southern Hemisphere 41=Lambert Conformal Conic, Secant cone, Northern Hemisphere 45=Lambert Conformal Conic, Secant cone, Southern Hemisphere 51=Cylindrical Equidistant 61=Northern Polar Azimuthal Equidistant 65=Southern Polar Azimuthal Equidistant	none
wedge_rotation	Int16	used for Lambert Conformal Conic, position of the empty wedge	1=up,2=left,3=down,4=right	none
mds_num	Int32	id number of mds, not used by NPOESS system	unitless	none
stan_lat1	Float64	first std latitude	radians/-pi/2 to +pi/2	none
stan_lat2	Float64	2 <sup>nd</sup> std latitude, needed for Lambert Conformal Conics,	radians/-pi/2 to +pi/2	none

Output	Type/Dimensions	Description	Units/Valid Range	Fill Value
		secant cone projections		
base_lon	Float64	base longitude of the X-Y coordinate system	radians/-pi to +pi	none
grid_inc_constant	Float64	determines scaling and grid sizes	unitless	none
grid_inc_c2	Float64	2 <sup>nd</sup> constant of same purpose, needed in cylindrical equidistant	unitless	none
grid_exponent	Float64	needed for Lambert projections	unitless	none
grid_constant_A	Float64	needed for Lambert projections	unitless	none
max_row	Float64	max grid row number	established at time MDS is created	none
max_col	Float64	max grid column number	established at time MDS is created	none
upr_left_lat	Float64	latitude of grid upper left corner	radians/-pi/2 to +pi/2	none
upr_left_lon	Float64	longitude of grid upper left corner	radians/-pi to +pi	none
lwr_right_lat	Float64	latitude of grid lower right corner	radians/-pi/2 to +pi/2	none
lwr_right_lon	Float64	longitude of grid lower right corner	radians/-pi to +pi	none
upr_left_x	Float64	x coordinate of upper left corner	established at time MDS is created	none
upr_left_y	Float64	y coordinate of upper left corner	established at time MDS is created	none
lwr_right_x	Float64	x coordinate of lower right corner	established at time MDS is created	none
lwr_right_y	Float64	y coordinate of lower right corner	established at time MDS is created	none
split_lon	Float64	longitude on both sides of empty sector for all lamberts	radians/-pi to +pi	none
costanlat	Float64	cosine tangent of latitude	unitless	none
latsml	Float64	smallest latitude in the MDS	radians/-pi/2 to +pi/2	none
latbig	Float64	largest latitude in the MDS	radians/-pi/2 to +pi/2	none
lonsml	Float64/[2]	smallest longitude in the MDS (need 2 because there might be 2 longitude ranges in the MDS, which happens when the MDS spans 180 degrees longitude)	radians/-pi to +pi	none
lonbig	Float64/[2]	biggest longitude in the MDS, need 2 for the same reason as lonsml	radians/-pi to +pi	none
num_lonrange	Int16	number of longitude ranges in MDS	always either 1 or 2	none

**Table 3.2.1.2.1-2. Gridded Geolocation Moderate Output (Geolocation)**

Output	Type/Dimensions	Description	Units/Valid Range	Fill Value
grow	Float64/ [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	row location in grid	unitless	ERR_FLOAT64_FILL
gcol	Float64/ [M_VIIRS_SDR_ROWS][M_VIIRS_SDR_COLS]	column location in grid	unitless	ERR_FLOAT64_FILL
ctr_grow	Float64/ [MOD_GEOLOC_ROWS][MOD_GEOLOC_COLS]	center coordinate of interpolation rectangle	unitless	ERR_FLOAT64_FILL
ctr_gcol	Float64/ [MOD_GEOLOC_ROWS][MOD_GEOLOC_COLS]	center coordinate of interpolation rectangle	unitless	ERR_FLOAT64_FILL
gmids	mids_type	map data set structure	See Table 3.2.1.2.1-3	See Table 3.2.1.2.1-3

**Table 3.2.1.2.1-3. Gridded Geolocation Imagery Output (Geolocation)**

Output	Type/Dimensions	Description	Units/Valid Range	Fill Value
grow	Float64/ [I_VIIRS_SDR_ROWS][I_VIIRS_SDR_COLS]	row location in grid	unitless	ERR_FLOAT64_FILL
gcol	Float64/ [I_VIIRS_SDR_ROWS][I_VIIRS_SDR_COLS]	column location in grid	unitless	ERR_FLOAT64_FILL
ctr_grow	Float64 [IMG_GEOLOC_ROWS][IMG_GEOLOC_COLS]	center coordinate of interpolation rectangle	unitless	ERR_FLOAT64_FILL
ctr_gcol	Float64/ [IMG_GEOLOC_ROWS][IMG_GEOLOC_COLS]	center coordinate of interpolation rectangle	unitless	ERR_FLOAT64_FILL
gmids	mids_type	map data set structure	See Table 3.2.1.2.1-3	See Table 3.2.1.2.1-3

**Table 3.2.1.2.1-4. Gridded Geolocation Day Night Band Output (Geolocation)**

Output	Type/Dimensions	Description	Units/Valid Range	Fill Value
grow	Float64/ [I_VIIRS_SDR_ROWS][I_VIIRS_SDR_COLS]	row location in grid	unitless	ERR_FLOAT64_FILL
gcol	Float64/ [I_VIIRS_SDR_ROWS][I_VIIRS_SDR_COLS]	column location in grid	unitless	ERR_FLOAT64_FILL
ctr_grow	Float64 [IMG_GEOLOC_ROWS][IMG_GEOLOC_COLS]	center coordinate of interpolation rectangle	unitless	ERR_FLOAT64_FILL
ctr_gcol	Float64/ [IMG_GEOLOC_ROWS][IMG_GEOLOC_COLS]	center coordinate	unitless	ERR_FLOAT64_FILL



Output	Type/Dimensions	Description	Units/Valid Range	Fill Value
		of interpolation rectangle		
gmids	mds_type	map data set structure	See Table 3.2.1.2.1-1	See Table 3.2.1.2.1-1

### 3.2.2 Algorithm Processing

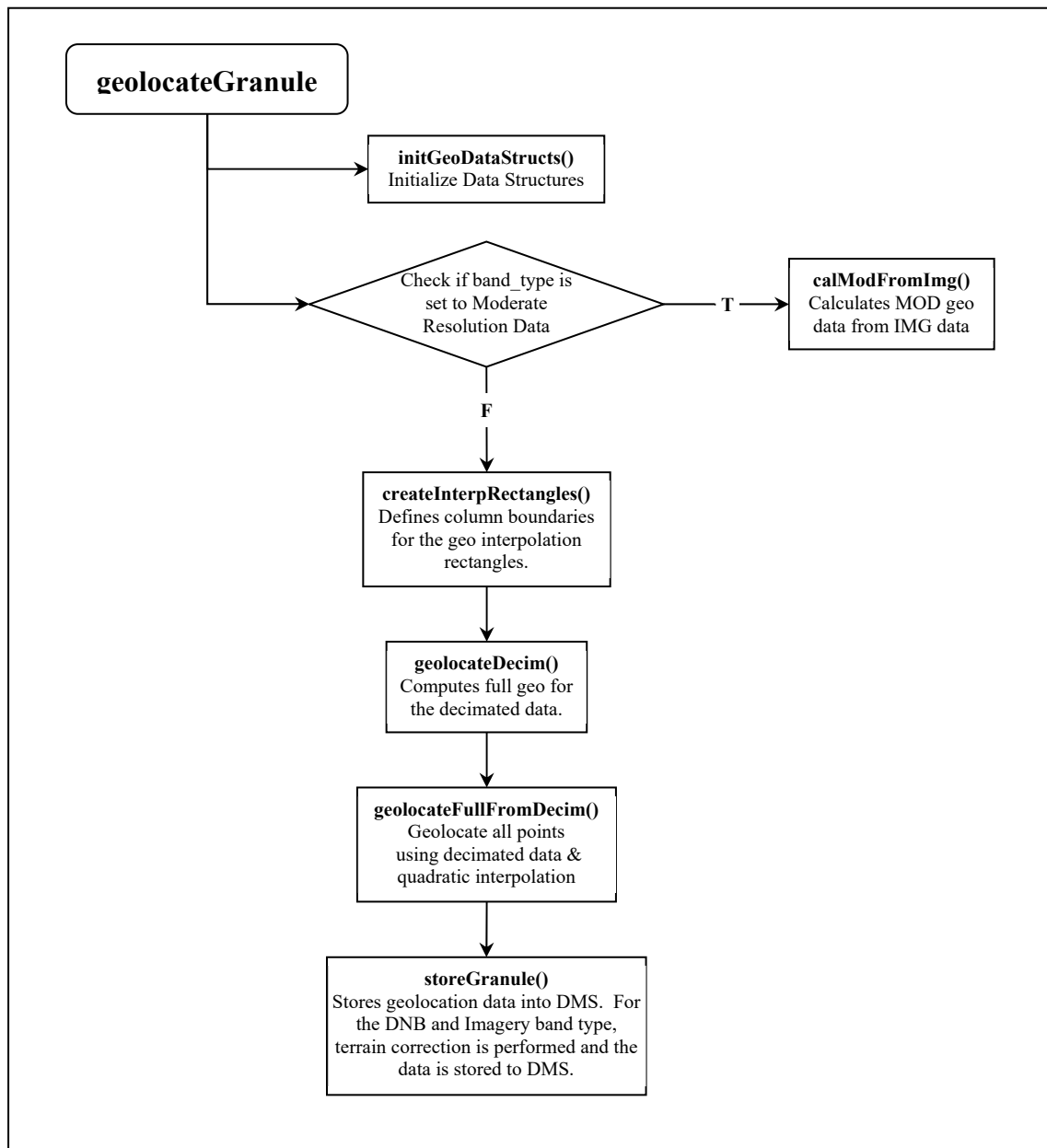
IDPS designed and developed a library of functions called common GEO that are used by all of the SDRs to calculate geodetic latitude and longitude as well as solar and lunar geometries. ProSdrCmnGeo methods are to: retrieve attitude and ephemeris information; retrieve moon vectors; determine the nadir latitude and longitude values of the scan; determine the ellipsoid intersect and satellite angles for each pixel and get terrain correction GEO values for each pixel.

Geolocation is performed using a combination of sensor specific functions and the Common Geolocation library of functions. The main driver for the geolocation algorithm is called `geolocateGranule()`. Consecutive calls to this driver are issued for each band type: DNB, Imagery, and Moderate. The Imagery Band (ellipsoid and terrain corrected) is processed first, followed by Moderate (ellipsoid and terrain corrected, and unaggregated Moderate), and finally the Day/Night Band (ellipsoid and terrain corrected). In order to meet latency requirements, the operational Geolocation code has implemented an interpolation scheme for computing ellipsoid geolocation. For DNB and Imagery resolution, ellipsoid geolocation is determined by performing full geolocation on a subset of pixels. The lat/long values for this subset of points are converted to a polar stereographic grid, and then quadratic interpolation is used to determine the geolocation for the remaining pixels. The results from the interpolation are then converted back to lat/long values. The terrain corrected product for DNB and Imagery is generated from the ellipsoid product for each lat/long pixel. The Moderate resolution ellipsoid product is created from the Imagery ellipsoid product by averaging four imagery pixels that surround each Moderate pixel. This averaging is done to compute the MOD ellipsoid geo for all pixels except for a subset of pixels near nadir, where full geolocation is performed (see Section 3.2.2.189 for more details). The resulting Moderate geolocation is then used to generate the terrain corrected product, as well as the unaggregated Moderate product. The unaggregated product is produced by a quadratic interpolation scheme that has been implemented as part of the code optimization for latency improvement.

#### 3.2.2.1 Main driver for Geolocation ( `geolocateGranule()` )

The main driver for geolocation is called `geolocateGranule()`. It calls the function `initGeoDataStructs()` to initialize the processing parameters, validate the scan encoder data, determine the thermal correction matrix, and to determine the band type dependent sample times. The next step in the `geolocateGranule()` function is to check the band type setting. If band type is set to Moderate resolution, then the function `calcModFromImg()` is called to calculate the Moderate geolocation products (ellipsoid, terrain corrected and unaggregated). If the band type is set to DNB or Imagery, the function `createInterpRctngls()` is called to define a subset of pixels based on rectangles that span across the scan and do not cross the aggregation zone boundaries. Next, `geolocateDecim()` is called to perform the full geolocation on the decimated set of points, and then `geolocateFullFromDecim()` is called to calculate the geolocation of the remaining pixels using quadratic interpolation. The final step of the `geolocateGranule()` function is to store the

geolocation data to memory using the function called `storeGranule()`. If the band type is DNB or Imagery, the terrain corrected product is produced in `storeGranule()`, and then saved to memory. For the ellipsoid DNB GEO, there are specific Terrain Correction fields that are copied from the DNB terrain correct product. The flow diagram for `geolocateGranule()` is shown in Figure 3.2.2.1-1.



**Figure 3.2.2.1-1. geolocateGranule() Flow Diagram**

### 3.2.2.2 GEO\_absolute\_limit\_check()

**GEO\_absolute\_limit\_check()** validates a set of input data samples against absolute limits. Flags are set to **BAD\_DATA** for all samples that are not within limits.

### 3.2.2.3 GEO\_determine\_DNB\_sample\_time\_offsets()

**GEO\_determine\_DNB\_sample\_time\_offsets()** determines the set of relative sample times for all pixels for the DNB. It uses aggregation zones defined by the **VIIRS-SDR-GEO-DNB-PARAM-LUT**, which provides lower and upper boundaries, aggregation mode index, and the number of

photosites per zone. The scan starts at the first aggregation zone and move through to the last zone at the end of scan.

The sample time determined is the time relative to the scan start. The scan start time is not added, as the GEO software already does this. The relative time is also used independently in several locations.

#### 3.2.2.4 GEO\_determine\_sample\_time\_offsets()

GEO\_determine\_sample\_time\_offsets() determines the set of relative sample times for all pixels for the requested resolution. It uses the sample index range to determine the aggregation zone and the number of samples (or observations) to be aggregated into a pixel.

The sample time for the first pixel is determined using the latch-to-center time and a time factor for the appropriate resolution. Subsequent sample times use the preceding sample time.

The sample time determined is the time relative to the scan start. The scan start time is not added, as the GEO software already does this. The relative time is also used independently in several locations.

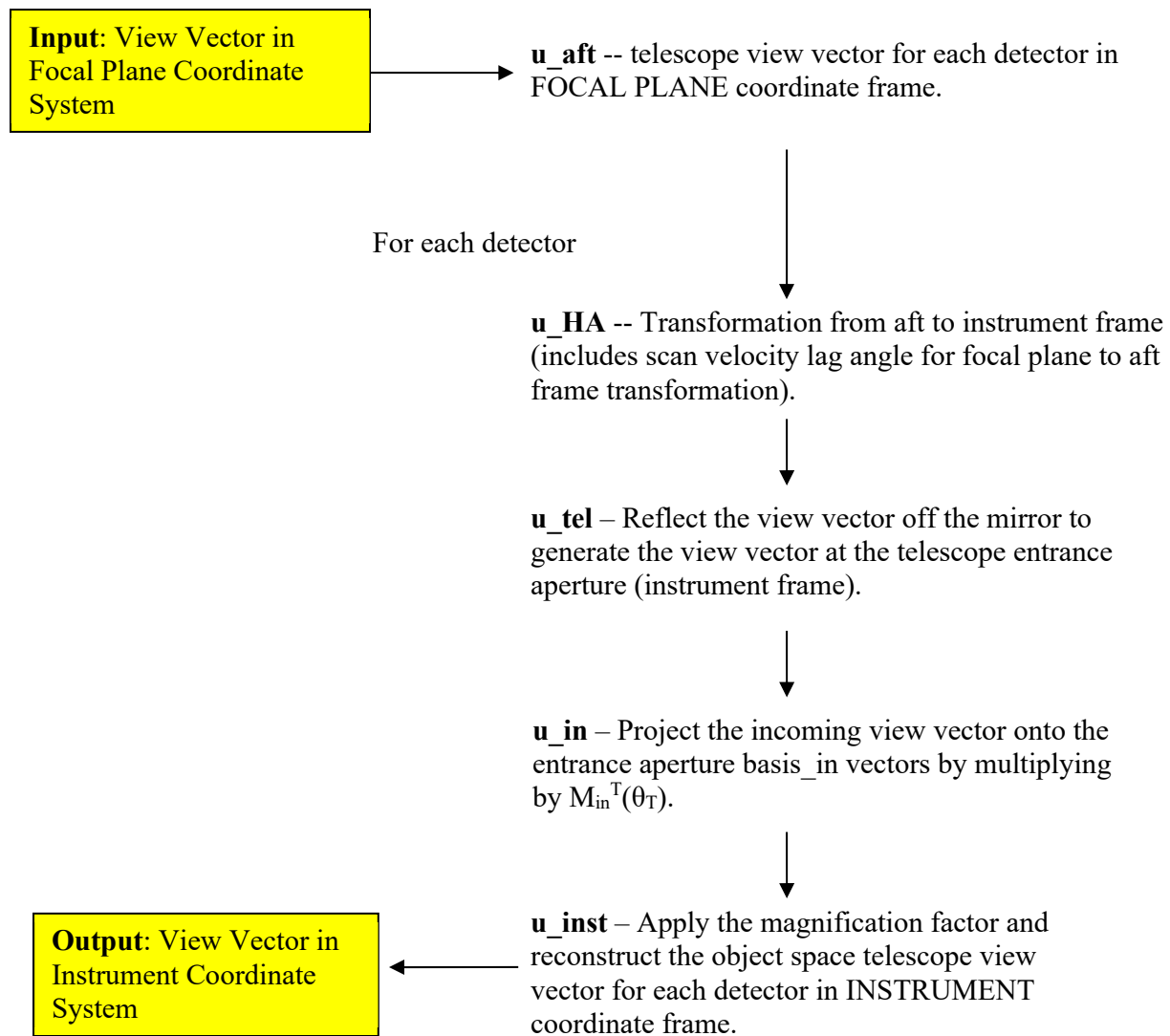
#### 3.2.2.5 GEO\_determine\_thermal\_corrections()

GEO\_determine\_thermal\_corrections() determines the temperature dependent corrections to apply to the granule's geolocation.

Note: The delivered routine of the science VIIRS geolocation code contained an untested or unverified version of temperature correction. All of the delivered test data had the routine 'turned off' by an indicator flag in the test data set. Rather than commenting out the sections of unverified code, it was determined to be better to replace the routine with a stubbed version that was always 'turned on' and returned zero correction for temperature. It is recognized that this routine is to be replaced once actual thermal correction testing verification has occurred. As the routine is currently implemented it should be easy to plug in a new routine into the existing code.

#### 3.2.2.6 GEO\_determine\_view\_vectors()

GEO\_determine\_view\_vectors() calculates object space view vectors in the instrument coordinate system for each along-track detector in a sample. See Figure 3.2.2.6-1 below.



**Figure 3.2.2.6-1. GEO\_determine\_view\_vectors() Block Diagram (Geolocation)**

### 3.2.2.7 GEO\_evaluate\_polynomial()

GEO\_evaluate\_polynomial() uses the input parameters to evaluate a polynomial.

### 3.2.2.8 GEO\_find\_next\_flag()

GEO\_find\_next\_flag() finds the next unflagged sample in a flag array.

### 3.2.2.9 GEO\_interp\_mod\_unagg()

GEO\_interp\_mod\_unagg() uses a quadratic interpolation method to calculate GEO from the values already calculated for the aggregated Moderate resolution, then stores per scan data into output structures based on resolution.

### 3.2.2.10 GEO\_interpolate\_mirror\_encoder()

For the samples that have mirror encoder data to interpolate between, GEO\_interpolate\_mirror\_encoder() interpolates the mirror encoder number using a linear interpolation method to interpolate the encoder data to the sample time, using the identified encoder times as bounding variables. Otherwise for DNB only, the function extrapolates the mirror encoder number using a linear extrapolation method to extrapolate the encoder data to the sample time. Extrapolation is not used for IMG and MOD bands.

### 3.2.2.11 GEO\_interpolate\_telescope\_encoder()

For the samples that have telescope encoder data to interpolate between, GEO\_interpolate\_telescope\_encoder() interpolates the telescope encoder number using a linear interpolation method to interpolate the encoder data to the sample time, using the encoder times as bounding variables. Otherwise for DNB only, the function extrapolates the telescope encoder number using a linear extrapolation method to extrapolate the encoder data to the sample time. Extrapolation is not used for IMG and MOD bands.

### 3.2.2.12 GEO\_process\_parameters()

GEO\_process\_parameters() processes input parameter files to extract key values used in GEO processing.

### 3.2.2.13 GEO\_relative\_limit\_check()

GEO\_relative limit check() validates a set of data samples by comparing differences between successive samples against a limit. Previously flagged values are not checked. Flags are set to BAD\_DATA for all samples not within limits.

### 3.2.2.14 GEO\_validate\_scan\_encoder\_data()

GEO\_validate\_scan\_encoder\_data() unpacks the telescope and mirror encoder data from the encoder and sector start segments. For each scan, it unpacks mirror encoder times from the encoder data, computes encoder values from the sector start and mirror side, and validates the data. Sensor model (instrument number in EDD) dependent TEL/HAM start encoder nominal values are used to determine if a sector rotation or sync loss condition occurs.

### 3.2.2.15 geolocatePixel()

This function performs full geolocation for a specific pixel. It uses the ProSdrCmnGeo class to calculate the ellipsoid intersection latitude, longitude and the azimuth, zenith angles to the spacecraft, sun, and moon.

### 3.2.2.16 geolocateDecim()

This function generates geolocation data for the subset of points that define the interpolation rectangles.

### 3.2.2.17 geolocateFullFromDecim()

This function uses quadratic interpolation to provide geolocation data for all pixels in the interpolation rectangles.

### 3.2.2.18 calcModFromImg()

This function calculates the geolocation data for the aggregated Moderate band type by averaging the four Imagery pixels that surround each Moderate pixel. The Imagery pixels are still in the default projection. Due to small satellite zenith values, the code performs full geolocation for pixels that are near nadir. This region is defined by pixel column (1531 through 1670 zero based). The function also generates the terrain corrected product from the ellipsoid for each lat/long pixel.

### 3.2.2.19 createInterpRctngls()

This function defines the start, middle, and end row/column numbers for each interpolation rectangle.

### 3.2.2.20 initGeoDataStructs()

This function initializes structures and setup data necessary for geolocation.

### 3.2.2.21 storeGranule()

This function copies the geolocation data for each band type into DMS. For the DNB and Imagery band types, terrain correction occurs here.

### 3.2.2.22 fixSatAngles()

This function corrects satellite azimuth angles. When the satellite zenith angle gets very small, the satellite azimuth angle changes rapidly. Because quadratic interpolation was used, the satellite azimuth angles need to be recalculated.

### 3.2.2.23 quadInterp()

This function performs quadratic interpolation.

### 3.2.2.24 geolocateAllRecPix()

This function performs full geolocation calculations for every pixel in an interpolation rectangle. These calculations are done only if an error does not occur in any of the nine points in the interpolation rectangle.

### 3.2.2.25 interpLocations()

This function interpolates the latitude and longitude values. The function converts the lat/long values to polar stereographic projection and performs the quadInterp function. The results of this are then converted back to lat/long values.

### 3.2.2.26 interpAngles()

This function interpolates the satellite, sun, and moon azimuth and zenith angles.

### 3.2.2.27 eclipseAndSaaFlags()

Used to determine solar eclipse, lunar eclipse and South Atlantic Anomaly quality flag values.

### 3.2.2.28 Geolocation Quality Flag Logic

Every output product has one pixel level quality flag. The pixel quality flag is an eight-bit field for each pixel indicating the status of the invalid input, bad pointing, bad terrain, invalid solar angles, and a spare unused bit. See Section 3.2.1.2. Any pixel level quality flag with a value of zero contains good data and a pixel level quality flag value other than zero contains anomalous data.

Each output also has two scan level quality flags. The first scan quality flag is an 8-bit field representing the Interpolation Stage, HAM impulse flag, SAA, Solar Eclipse, Lunar Eclipse flags and HAM side. The second quality flag is an 8-bit field representing the Scan Encoder Electronics side. See Section 3.2.1.2. Details of the logic used to set these quality flags can be learned by examining the geolocation functions along with the `geolocatePixel()` and `geolocateGranule()` functions.

### 3.2.2.29 Create Degree GEO Products

After geolocation and calibration processing have been completed then the delivered GEO products, with angles in degrees, are created by calling `createDegreeProducts()` and `convertToDegrees()`. This process is performed for all six non-gridded geolocation products.

Angle-values such as latitude, longitude, and satellite and solar azimuth and zenith angles are stored as radians for the internal GEO products and as degrees for the delivered GEO products. The DNB output GEO has no corresponding radian (internal) form, and reports all of the angle-values previously mentioned, with the addition of lunar azimuth and zenith angles and moon phase, in degrees.

## 3.2.3 Graceful Degradation

The VIIRS SDR Geolocation procedure contains no graceful degradation.

### 3.2.3.1 Graceful Degradation Inputs

None.

### 3.2.3.2 Graceful Degradation Processing

None.

### 3.2.3.3 Graceful Degradation Outputs

None.

## 3.2.4 Exception Handling

The error handling concept used produces a product if at all possible. This is accomplished by using error fill values to fill data structures at the scan and pixel levels, setting flags and continuing with processing to recover from minor errors and still be able to produce a product. Error handling has been implemented at the granule, scan, and pixel levels of the code.

At the granule level, invalid function arguments, out-of-bound arrays, and failed function calls result in error messages being sent via the INF log mechanism and exiting the function by returning a fail value (with the exception of a failed call to `GEO_get_Grinding_points()` which



causes an error message to be sent and fill values to be used). When this occurs, no GEO product for that granule is produced. If missing scans are detected, an error message is sent, the scans are filled with fill values, and processing continues.

At the scan level, invalid function arguments and out-of-bound arrays result in error messages being sent via the infrastructure log mechanism and exiting the function by returning a fail value to the granule level function. For invalid IET time, invalid satellite attitude and ephemeris, or invalid mirror side, a flag is set, the entire scan is filled with the appropriate fill values and processing continues with the next scan. For invalid satellite ephemeris and attitude data, invalid sun or moon angles, no ellipse intersect, failed terrain correction, or failure in determining view vectors, a flag is set, the sample is filled with the appropriate fill values, and processing continues with the next sample in the scan.

At the pixel level, invalid function arguments, out-of-bound arrays, and failed function calls result in error messages being sent via the infrastructure log mechanism and exiting the function by returning a fail value. The returned “fail values” are caught by the scan level calling functions resulting in the pixel being filled with the error fill value, a flag being set, and processing continuing with the next pixel.

### 3.2.5 Data Quality Monitoring

No data quality monitoring is performed by the VIIRS SDR Geolocation procedure.

### 3.2.6 Computational Precision Requirements

All internal calculations are done at double precision, and they need to be done this way as small scan angle errors lead to large GEO errors. The geodetic Latitude and Longitude fields are output as floats, which can introduce an uncertainty no larger than two meters in Earth location. Sensor, sun and lunar angles are output as floats.

### 3.2.7 Algorithm Support Considerations

The DMS and INF must be running before the algorithm is executed.

### 3.2.8 Assumptions and Limitations

## 3.3 VIIRS SDR Calibration Description

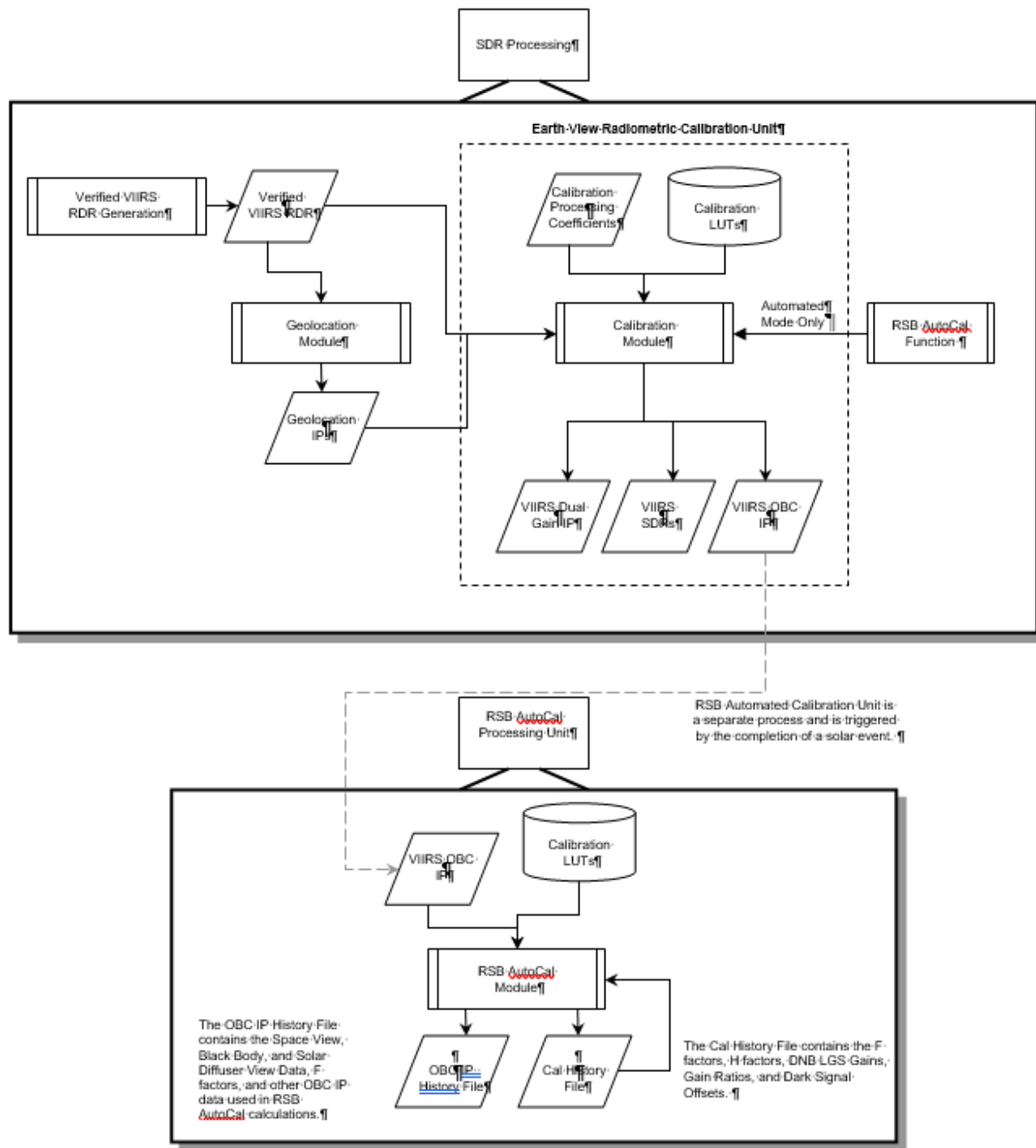
The purpose of the Radiometric CAL Component is to convert the VIIRS digital counts to radiance fields at the top of atmosphere (TOA), as well as brightness temperature and reflectance values. The algorithm used for radiometric CAL has been derived from first principles and then simplified through judicious specifications of algorithm input parameters. The Radiometric CAL Component processes the Verified VIIRS Raw Data Records (RDRs) to produce VIIRS SDRs, VIIRS On-board Calibrator (OBC) Intermediate Product (IP), and VIIRS Calibrated Dual Gain IP<sup>1</sup>. The output SDR data fields include TOA radiances, reflectances, and brightness temperature fields, in addition to quality information. The science code consists of two independent, executable units: Earth View (EV) Radiometric CAL Unit and a Reflective Solar

---

<sup>1</sup> For dual gain bands only: M1 – M5, M7 and M13.

---

Band Automated Calibration (RSB AutoCal) Processing Unit. The EV Radiometric CAL Unit is executed on a near real-time basis, whereas the RSB AutoCal Processing Unit is executed once per orbit. Outputs from the EV Radiometric CAL Unit are the VIIRS SDRs for each resolution, the Dual Gain IP, and the OBC IP. The RSB AutoCal Processing Unit produces scale factors,  $F$ , that are used as a multiplier to the CAL lookup tables (LUTs) used by the EV Radiometric CAL Unit. Refer to Section 3.3.1.2 for details on output SDRs and IPs. The Calibration Processing Chain is shown in Figure 3.3-1.



**Figure 3.3-1. Calibration Processing Chain**

### 3.3.1 Interfaces

#### 3.3.1.1 Inputs

The inputs to the Earth View Radiometric Calibration Unit of the VIIRS Radiometric CAL Component are the Verified VIIRS RDRs, the VIIRS GEO IPs, and a suite of files that include LUTs, CAL coefficient LUTs and CAL run-time parameter LUTs. The input files are

summarized in 474-00448-01-06\_JPSS-SRS-Vol-I-Part-06, Table 3-1 (rows 4-6, 9, 12, 15, 18-19). Table 3.3.1.1-1, below, provides a brief description and reference to the Data Dictionary (if there is one) of each input.

**Table 3.3.1.1-1. Earth View Radiometric Calibration Unit Inputs**

Input	Description	Reference Document
VIIRS Verified RDR	Data containing digital numbers for earth view, onboard calibrator view and space view data, as well as engineering data, health and safety data. This is raw satellite data that has been reformatted, pre-processed, and verified prior to processing. This is a heap item.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Geolocation Products	Data containing terrain corrected solar zenith and azimuth angles, satellite zenith and azimuth angles, as well as latitudes and longitudes for each VIIRS grid point for each of the three VIIRS resolutions. (375m, 750m, and DNB).	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Geolocation OBC-IP	The VIIRS On Board Calibrator IP file contains solar diffuser observations, the associated gain state and HAM side information, and all engineering and housekeeping data, including unscaled data from the Solar Diffuser Stability Monitor (SDSM)/VIIRS Earth View Radiometric Calibration Unit and the Solar Diffuser GEO angles	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS SDR Calibration LUTs	Look up tables required for VIIRS SDR Calibration processing/ Refer to 474-00448-02-06_JPSS-DD-Vol-II-Part-06 for a specific*** LUT.	474-00448-02-06_JPSS-DD-Vol-II-Part-06

\*\*\*Several of the LUT input items contain platform dependent values. Each platform must have a separate LUT associated with that specific sensor. During the input phase of the algorithms IPO processing model, the spacecraft ID provided in the tasking information is used as search metadata in order to ensure the correct LUT is retrieved from DMS.

### 3.3.1.2 Outputs

The VIIRS EV Radiometric CAL Unit produces 22 SDRs and two IPs, as shown in 474-00448-01-06\_JPSS-SRS-Vol-I-Part-06, Table 3-1. A separate SDR is output for each band in each resolution: Imagery (I1-I5), Moderate (M1-M16), and DNB. All Imagery resolution SDRs and Moderate resolution SDRs contain the full resolution geolocated calibrated TOA radiances. In addition, reflective band SDRs (bands I1-I3 and M1-M11) contain full resolution calibrated reflectances and emissive band SDRs (bands I4-I5 and M12-M16) contain full resolution calibrated TOA brightness temperatures. The DNB SDR contains the full resolution geolocated

calibrated TOA radiances for the DNB. The OBC IP contains sensor calibrator view data (space view, blackbody, and solar diffuser) from all VIIRS bands arranged by scan and unprocessed engineering and housekeeping data. The OBC IP is the input to the Reflective Solar Band Automated Calibration Processing Unit and is also the primary input to offline Cal/Val analysis. The Calibrated Dual Gain IP contains unaggregated TOA radiances for each dual gain band for the nadir (aggregation by three) and the near-nadir (aggregation by two) aggregation zones.

**Table 3.3.1.2-1. Earth View Radiometric Calibration Unit Outputs**

Input	Description	Reference Document
VIIRS Mod Band 1-16 SDR	Band M1 - radiance & reflectance at nominal center wavelength 412 nm. Band M2 - radiance & reflectance at nominal center wavelength 445 nm. Band M3 - radiance & reflectance at nominal center wavelength 488 nm. Band M4 - radiance & reflectance at nominal center wavelength 555 nm. Band M5 - radiance & reflectance at nominal center wavelength 672 nm. Band M6 - radiance & reflectance at nominal center wavelength 746 nm. Band M7 - radiance & reflectance at nominal center wavelength 865 nm. Band M8 - radiance & reflectance at nominal center wavelength 1240 nm. Band M9 - radiance & reflectance at nominal center wavelength 1378 nm. Band M10 - radiance & reflectance at nominal center wavelength 1610 nm. Band M11 - radiance & reflectance at nominal center wavelength 2250 nm. Band M12 - radiance & emittance at nominal center wavelength 3700 nm. Band M13 - radiance & emittance at nominal center wavelength 4050 nm. Band M14 - radiance & emittance at nominal center wavelength 8550 nm. Band M15 - radiance & emittance at nominal center wavelength 10763 nm. Band M16 - radiance & emittance at nominal center wavelength 12013 nm.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Cal DQTT	Image bands 1-5, Moderate bands 1-16 and DNB Data Quality Test Tables	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Img Band 1-5 SDR	Band I1 - radiance & reflectance at nominal center wavelength 640 nm. Band I2 - radiance & reflectance at nominal center wavelength 865 nm. Band I3 - radiance & reflectance at nominal center wavelength 1610 nm. Band I4 - radiance & reflectance at nominal center wavelength 3740 nm. Band I5 - radiance & reflectance at nominal center wavelength 11450 nm	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS DNB SDR	VIIRS DNB measures radiance over a panchromatic band at wavelengths between 500 nm and 900 nm.	474-00448-02-06_JPSS-DD-Vol-II-Part-06

Input	Description	Reference Document
VIIRS Dual Gain Calibrator SDR	The VIIRS calibrated dual-gain band IP includes unaggregated top-of-atmosphere radiance values for the dual-gain spectral bands in their nadir (3:1) and near-nadir (2:1) aggregation zones. The included radiance values from the moderate resolution dual-gain bands are identified as: M1, M2, M3, M4, M5, M7, and M13.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS OBC IP	The VIIRS Onboard Calibrator IP contains subsets of the Verified VIIRS RDR and is the primary input to offline VIIRS Cal/Val performance analysis.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Cal DQN	Imagery bands 1-5, Moderate bands and DBN Data Quality Notifications	474-00448-02-01_JPSS-DD-Vol-II-Part-1

### 3.3.1.2.1 Definitions of the Calibration VIIRS SDR Outputs

The contents of the VIIRS SDR outputs are described in 474-00448-02-06\_JPSS-DD-Vol-II-Part-06 (specific sections are prescribed in the table above for each SDR type. The VIIRS SDR CAL routine outputs two structures for each band in the Moderate (750m) resolution, and Imagery (375m) resolution: a scaled version and an unscaled version of the SDR. The scaled versions of the bands are identified for long term storage, and the non-scaled full floating point versions are used for immediate processing of EDRs and IPs. These full floating point versions are removed by the storage system after an amount of time, post creation. Quality Flags (Scan, Pixel, RDR Scan, Reduced Quality and Bad Detector) are also found in the product profiles for each band.

### 3.3.1.2.2 VIIRS On-Board Calibrator Intermediate Product

474-00448-02-06\_JPSS-DD-Vol-II-Part-06, Section 8.2 summarizes the contents of the VIIRS OBC IP output. The OBC IP contains the same granule metadata as the output SDR.

## 3.3.2 Algorithm Processing

This is the derived algorithm for the Earth View Radiometric CAL Unit algorithm and is a subclass of the AutoGeneratedProSdrViirsCal and ProCmnAlgorithm classes. The derived algorithm class creates a list of input data items read from DMS and passes required data into the algorithm. All output data items are written to DMS once the algorithm finishes processing this data.

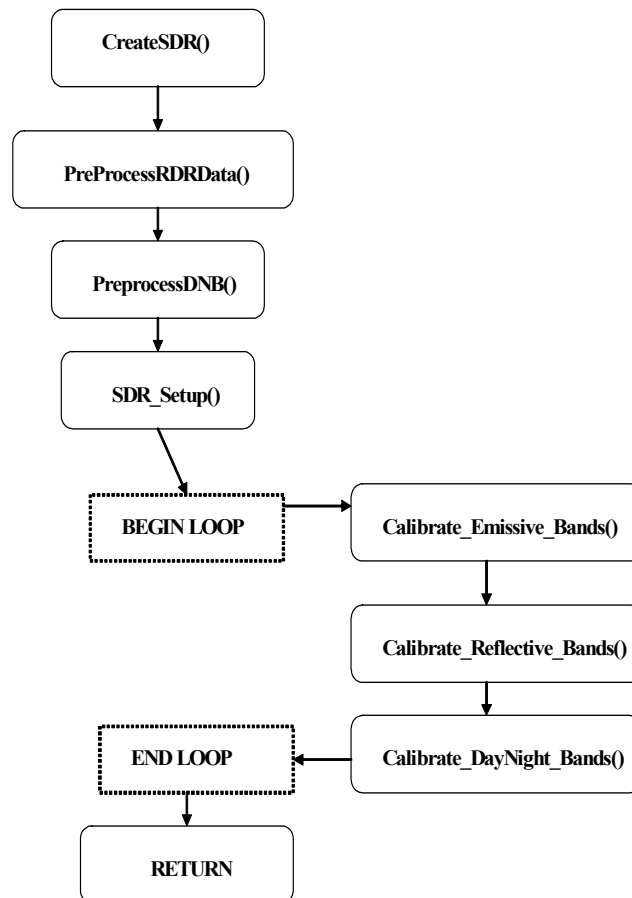
The objective of the Radiometric CAL algorithm is to produce TOA radiances, reflectances (reflective bands only), and brightness temperatures (emissive bands only) for each pixel in an imagery, ideal moderate or day/night band. The radiometric CAL equations are derived from first principles. The equations that are implemented in the code have been simplified through extensive modeling and analysis and careful specification of the algorithm input parameters.

The Earth View Radiometric CAL Unit plus functions for the Reflective Solar Band Automated Calibration Processing Unit are described functionally in the following sections. Figure 3.3.2-1 shows the Earth View Radiometric Calibration Flow. Figure 3.4.2.1-1. shows the Reflective Solar Band Automated Calibration processing flow. For details on the structure of the various functions, please refer to embedded code comments and the function descriptions located in

Section 3.2.2. For variable and structure definitions please refer to embedded code comments and the data dictionary in Rose.

The logic flow of the main program for the Earth View Radiometric CAL algorithm is provided in Figure 3.3.2-1. The core of the CAL algorithm occurs in the functions:

`Calibrate_Emissive_Bands()`, `Calibrate_Reflective_Bands()`, and `Calibrate_DayNight_Band()`. Core equations for the reflective and emissive radiometric CAL algorithms are specified in Table 3.2.1.2.1-2, Table 3.2.1.2.1-3, and Table 3.2.1.2.1-4. A summary of the symbols and units used in these equations can be found in Table 3.2.1.2.1-1.



**Figure 3.3.2-1. Radiometric Calibration Main Program Flow**

### 3.3.2.1 Sub-frame Offset Compensation

It has been determined that in imagery band samples there is a consistent offset difference between the 1<sup>st</sup> (odd) and 2<sup>nd</sup> sub-frames (even). This anomaly is apparent in both space view and black body samples. In order to compensate for this difference, averages are computed for each sub-frame of each offset type, space view or black body. These subframe averages are then swapped per even/odd detector according to the `swapEvenOdd` arrays and are then used in calibration. Note that this swapping does not occur for the data written out to the OBC IP.

### 3.3.2.2 Robust Algorithm Design

Each scan contains data from four views: earth, solar diffuser, space, and black body. In addition each scan contains data regarding DC restore voltage for each detector and thermistor temperatures. These data are used in the calibration algorithm. Since calibration is dependent on all of these inputs, when any input is invalid then an alternate data source is used to replace the invalid data and calibration continues. Scans calibrated using an alternate data source are indicated by a non-zero value in the SDR reduced quality flag field.

### 3.3.2.3 Earth View Radiometric Calibration Error Handling

When the radiometric metrics cannot be computed due to missing input data, dead detectors, invalid electronic background digital numbers, invalid computed CAL coefficients, saturated earth view pixels, or a solar zenith angle greater than 89 degrees (reflectance values only) then the pixel quality flag is set to the appropriate value. The dual gain band pixel quality flags are set according to the convention outlined in 474-00448-02-06\_JPSS-DD-Vol-II-Part-06, Section 8.3.

### 3.3.2.4 calibrateSDR()

This is the main driver for the science processing of the EV Radiometric CAL algorithm. It controls each of the calls to the separate modules described below, and is responsible for handling stop calls and processing failures.

### 3.3.2.5 Preprocess\_RDR\_Data() and Preprocess\_DNB()

These routines are pre-process parameters needed for scan by scan calibration. Temperatures are determined from VIIRS engineering data. These functions calculate frame-averaged SV counts and standard deviation with outlier rejection for reflective bands. Calibrator gain states are extracted and telemetry and calibrator data are copied to the OBC\_IP output item.

### 3.3.2.6 SDRSetup()

This routine performs a variety of functions in preparation for EV calibration. It calculates radiance and reflectance coefficients, sets up data structures used by CAL routines, as well as initializes QA parameters.

### 3.3.2.7 Combine\_Reduced\_Quality\_Flags()

This routine calculates and stores the SDR reduced quality flag data in the SDR.

### 3.3.2.8 Calibrate\_Emissive\_Bands()

This function calculates emissive band calibrated TOA radiance and brightness temperature for a scan.

### 3.3.2.9 Find\_SV\_Scan\_Index()

When a dual gain band undergoes either emissive or reflective calibration, the space view with corresponding mirror side and gain state to the current pixel must be used to correctly calibrate the dn value. Due to a dcr state change every four scans, only the space view with the corresponding parameters that is also within the dcr grouping should be used for calibration. At the beginning and end of a granule, it is possible the space view scan required is located within



---

the previous or following granule. This routine determines the correct granule and scan number to use for accurate dual gain calibration.

#### 3.3.2.10 Calibrate\_Reflective\_Bands()

This function calculates reflective band calibrated TOA radiance and reflectance for a scan.

#### 3.3.2.11 Calibrate\_DayNight\_Bands()

This function calculates DNB calibrated TOA radiance for a scan. The first frame of a scan line for each of the 16 DNB detectors is filled with zero integers for the Earth View and the three calibration views (Space View, Blackbody View and Solar Diffuser View) in the RDR. This process of zeroing out the first frame for the DNB scan lines is implemented for sensor FU1 to fix an anomaly with hardware.

#### 3.3.2.12 Aggregate\_750M\_DG()

This function aggregates the dual bands into the single gain resolution. It is responsible for storing the aggregated values in the SDR outputs and the unaggregated values in the Dual Gain IP product.

#### 3.3.2.13 viirs\_decomp()

This code decompresses VIIRS earth view packet data. See code headers for copyright and redistribution restrictions.

### 3.3.3 Graceful Degradation

The VIIRS SDR Calibration procedure contains no graceful degradation.

#### 3.3.3.1 Graceful Degradation Inputs

None.

#### 3.3.3.2 Graceful Degradation Processing

None.

#### 3.3.3.3 Graceful Degradation Outputs

None.

### 3.3.4 Exception Handling

The code primarily checks for invalid reference values for input and output structures, division by epsilon, and invalid array indexes. These conditions are handled on a case by case basis depending on the purpose of the function they are located within. In all cases an error or debug message is logged. For the dual-gain bands, multiple RDR pixels are aggregated (two or three) together to produce a single output SDR pixel. This creates the possibility of an SDR output pixel where there is a mix of good and bad quality RDR pixels. The dual-gain SDR products therefore have a special quality flag that indicates an aggregated mix of good and bad quality identified as NON\_NOMINAL (see 474-00448-02-06 JPSS-DD-Vol-II-Part-06, Section 8.3). Problems with CAL cause quality flags to be applied to all SDR pixels that derive from the CAL computation. If CAL source data is missing (space view, solar diffuser, or OBC BB) then a CAL

---

exception occurs and this is applied as a quality flag to all earth view SDR output pixels that require that calibration. Since CAL applies over a whole scan or over multiple scans, this causes a quality flag to be issued for the entire scan.

### 3.3.5 Data Quality Monitoring

474-00448-02-06\_JPSS-DD-Vol-II-Part-06, (Sections referencing HDF5 Metadata Details), show the VIIRS SDR Calibration data quality monitoring.

### 3.3.6 Computational Precision Requirements

The VIIRS SDR CAL routine outputs two structures for each band in the Moderate (750m) resolution, and Imagery (375m) resolution: a scaled version and a non-scaled version of the SDR. The scaled versions of the bands are identified for long term storage, and the non-scaled full floating point versions are used for immediate processing of EDRs and IPs. These full floating point versions are removed by the storage system after an amount of time, post creation.

### 3.3.7 Algorithm Support Considerations

The DMS and INF must be running before the algorithm is executed.

### 3.3.8 Assumptions and Limitations

#### 3.3.8.1 I/O Timeliness

Dual gain bands require calibration space view data from the previous granule and next granule. From this, one-granule latency occurs for the dual gain bands.

#### 3.3.8.2 Sci2Ops Issues

The CAL code expects that the Verified VIIRS RDR data follows the “product order” convention for detector numbering, which requires that the highest number detector in scan N lies adjacent to detector number 1 in scan N+1.

#### 3.3.8.3 Additional Assumptions

It is assumed that Verified RDR data are provided in correct time sequence and that complete scans are provided (missing data within a scan is to be filled). It is also assumed that RDR data are reconstructed to remove any coding, compression, and band referencing that was performed on-board.

It is assumed that the time and spacecraft parameters provided in the downlinked data and extracted by the VIIRS Build RDR module are present and correct unless a quality flag is set.

It is assumed that the VIIRS GEO IPs are available at the time of execution of the science code.

It is assumed that sufficient resources are available to support operational processing using the VIIRS Build SDR radiometric CAL algorithm.

It is assumed that the required coefficients and look up tables are complete and provided in the correct format.

## 3.3.8.4 Additional Limitations

None.

**3.4 VIIRS SDR Reflective Solar Band (RSB) Automated Calibration (RSBAutoCal) Description**

## 3.4.1 Interfaces

## 3.4.1.1 Inputs

The inputs for the Reflective Solar Band Automated Calibration (RSBAutoCal) Data Processing Unit are the VIIRS OBC IP and a suite of Calibration parameter files that include LUTs and CAL coefficients and Auxiliary (AUX) files. The input files are summarized in 474-00448-01-06\_JPSS-SRS-Vol-I-Part-06, Table 3-1 (rows 13-14, 24) and below:

**Table 3.4.1.1-1. VIIRS SDR RSBAutoCal Inputs**

Input	Description	Reference Document
VIIRS-RSBAUTOCAL-OBCIP-HISTORY-AUX	Contains the RSB F table along with a collection of information for VIIRS-OBC-IPs that were processed in the past.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS-RSBAUTOCAL-HISTORY-AUX	The VIIRS Reflective Solar Band (RSB) Automatic Calibration (AutoCal) History contains the last 20 updates of the RSB, F, H, DNB LGS gain, DNB dark Signal, DNB gain ratios, and solar processing data	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS-OBC-IP	The VIIRS Onboard Calibrator IP contains subsets of the Verified VIIRS RDR and is the primary input to offline VIIRS Cal/Val performance analysis.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS RSBAutocal LUTs	Look up tables required for VIIRS RSBAutoCal processing. Refer to 474-00448-02-06_JPSS-DD-Vol-II-Part-06 for a specific LUT.	474-00448-02-06_JPSS-DD-Vol-II-Part-06

## 3.4.1.2 Outputs

The RSBAutoCal Data Processing Unit has two output files, as described below.

**Table 3.4.1.2-1. VIIRS SDR RSBAutoCal Outputs**

Input	Description	Reference Document
VIIRS-RSBAUTOCAL-OBCIP-HISTORY-AUX	Contains the RSB F table along with a collection of information for VIIRS-OBC-IPs that were processed in the past.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS-RSBAUTOCAL-HISTORY-AUX	The VIIRS Reflective Solar Band (RSB) Automatic Calibration (AutoCal) History contains the last 20 updates of the RSB, F, H, DNB LGS gain, DNB dark Signal, DNB gain ratios, and solar processing data	474-00448-02-06_JPSS-DD-Vol-II-Part-06

The OBCIP History file contains the RSB F table as well as a collection of useful information from OBCIPs processed. The Cal History file contains the last 20 updates of the RSB F, H, DNB LGS gain, DNB dark signal, DNB gain ratios, and solar processing data.

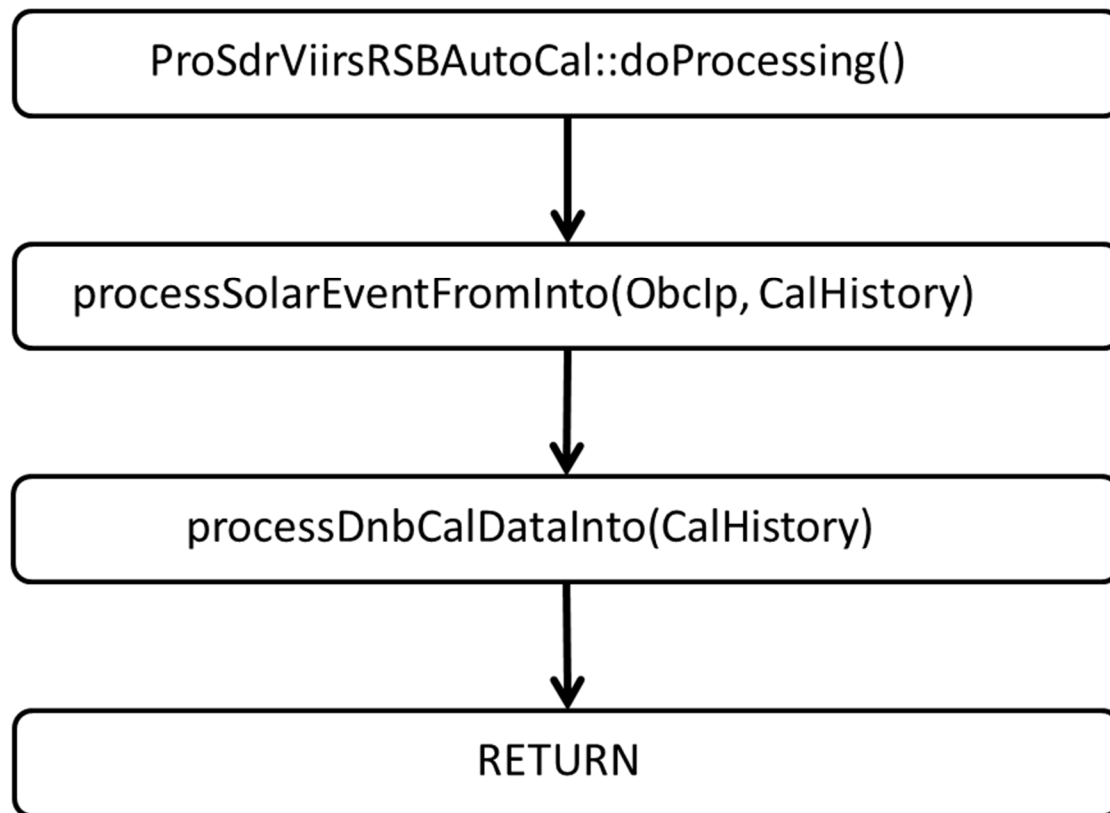
### 3.4.2 Algorithm Processing

This is the derived algorithm for the RSBAutoCal Unit algorithm and is a subclass of the ProCmnAlgorithm class. The derived algorithm class creates a list of input data items read from DMS and passes required data into the algorithm. All output data items are written to DMS once the algorithm finishes processing this data.

#### 3.4.2.1 Automated Reflective Solar Band Calibration Data Processing Logic

The logic flow of the main program for the RSBAutoCal Data Processing algorithm is provided in Figure 3.4.2.1-1. The core of the RSBAutoCal Data occurs inside processSolarEventFromInto() and processDnbCalDataInto(CalHistory), where RSB F, H, DNB LGS gain, DNB dark signal, and DNB gain ratios are calculated. The outputs are stored in Cal History file.

Core equations used for the Automated Reflective Solar Band Calibration Data Processing algorithm are specified in Table 3.4.2.3-1.



**Figure 3.4.2.1-1. High level RSBAutoCal Data Processing Main Program Flow**

## 3.4.2.2 Calibration Symbols and Units

**Table 3.4.2.2-1. Calibration Symbols and Units**

Symbol	Un-notated Indices <sup>2</sup>	Description	Units
$a_1(T_{det})$	B, d, g, m	Temperature dependent first order (linear) coefficient of the response function of a detector circuit. This is the effective capacitance of the detector circuit.	Photoelectrons/V
$a_2(T_{det})$	B, d, g, m	Temperature dependent second order coefficient of the response function of a detector circuit.	Photoelectrons/V <sup>2</sup>
$agg(N_F)$	N/A	Along-scan aggregation zone for DNB.	Unitless
$B$	N/A	Band number.	Unitless
$b_1(T_{elec})$	B, d, m	Temperature dependent first order (linear) coefficient of the response function of electronics. This is the inverse of the gain of the combined ADC & ASP circuits.	V/count
$b_2(T_{elec})$	B, d, m	Temperature dependent second order coefficient of the response function of electronics.	V/count <sup>2</sup>
$BRDF(\phi_h, \phi_v, \lambda, t)$	B, d	Bi-directional reflectance distribution function of solar diffuser expressed in terms of lab angles.	1/sr
$c_0$	B, d, g, m	0 order coefficient of the radiance response function.	W/(m <sup>2</sup> μm sr)
$c_1(T_{det}, T_{elec})$	B, d, g, m	Temperature dependent first order (linear) coefficient of the response function for radiance.	W/(m <sup>2</sup> μm sr cnts)
$c_2(T_{det}, T_{elec})$	B, d, g, m	Temperature dependent second order coefficient of the response function for radiance.	W/(m <sup>2</sup> μm sr cnts <sup>2</sup> )
$c_j(T_{det}, T_{elec})$	B, d, g, m	Temperature dependent jth order coefficient of the response function after calibration update.	W/(m <sup>2</sup> μm sr cnts <sup>j</sup> )
$c[agg(N_F), N_P, N_G]$		Pre-determined DNB calibration coefficients, dependent on $N_F$ , $N_P$ , and $N_G$ .	W/(m <sup>2</sup> μm sr cnts <sup>i</sup> )
$\Delta c_j$	B, d, g, m	Adjustment to coefficient of the response function.	W/(m <sup>2</sup> μm sr cnts <sup>j</sup> )
$d_j(T_{det}, T_{elec})$	B, d, g, m	Temperature dependent jth order coefficient of the response function for reflectance.	1/counts <sup>j</sup>
$d_{se}(t)$		Distance from sun to earth at time $t$ .	Meter
$\overline{d_{se}}$		Distance from sun to earth averaged over a year.	Meter
$dn_{sd}(t)$	B, d, g, m, n	Differential detector counts at solar diffuser with space view subtracted.	Counts
$\overline{dn_{sd}(t_n)}$	B, d, g, m	Differential detector counts at solar diffuser averaged over the acquisition at time $t_n$ .	Counts
$dn_{obc}(t)$	B, d, g, m, n	Differential detector counts at OBCBB with space view subtracted at time $t$ .	Counts
$\overline{dn_{obc}(t_n)}$	B, d, g, m	Differential detector counts at OBCBB averaged over the acquisition at time $t_n$ .	Counts
$DN$	B, d, g, m, n	Total detector counts.	Counts
$DN_{DNB}$		Total DNB detector counts.	Counts
$DN_{SV\_DNB}[N_F, N_P, N_G]$		DNB space view offset LUT, as a function of $N_F$ , $N_P$ , and $N_G$ .	Counts
$dn_{ev}$	B, d, g, m, n	Differential detector earth view counts with space view subtracted.	Counts
$dn_{DNB}$		Differential DNB detector counts with space view subtracted.	Counts

<sup>2</sup> In order to reduce complexity of the notation some indices are dropped in the equations in this document. The following abbreviations are used in this column. B=band number; d=detector number; g=gain state; m=mirror side; n=cross-track pixel number

Symbol	Un-notated Indices <sup>2</sup>	Description	Units
$\overline{DN}_{sv}$	B, d, g, m	Total detector counts per frame averaged over space view.	Counts
$E_{sun}(\lambda, t), E_{sun}(\lambda, d_{se})$		Irradiance from the sun upon a surface with normal pointing toward the sun.	W/(m <sup>2</sup> μm)
$F$	B, d, g, m	Factor for update of the radiance coefficients.	Unitless
$F_{cav}$	B	Factor describing the effective solid angle of the cavity as seen by the OBD BB.	sr
$F_{sh}$	B	Factor describing the effective solid angle of the shield as seen by the OBD BB.	sr
$F_{tele}$	B	Factor describing the effective solid angle of the telescope as seen by the OBD BB.	sr
$G$	B	Gain converting detector electron counts to radiance.	W/(m <sup>2</sup> μm sr) /photoelectron
$\overline{\Delta L}_{det}(\theta, B)$	d, m	Differential band-averaged detected spectral radiance at field stop for angle $\theta$ relative to space view.	W/(m <sup>2</sup> μm sr)
$\overline{L}_{ap}(\theta, B)$	d	Band-averaged spectral radiance at the aperture for scan angle $\theta$ .	W/(m <sup>2</sup> μm sr)
$\overline{L}_{nsd}(\phi_h, \phi_v, B)$	d	Band-averaged normalized solar diffuser spectral radiance.	W/(m <sup>2</sup> μm sr)
$\overline{L}_{DNB}$		Band-averaged spectral radiance at the aperture for Day Night Band.	W/(m <sup>2</sup> μm sr)
$L(T, \lambda)$		Blackbody spectral radiance according to Planck's function.	W/(m <sup>2</sup> μm sr)
$L_{obc\_rft}(T_{sh}, T_{cav}, T_{tele}, \lambda)$	B, d, n	Spectral radiance emissive background from shield, cavity and telescope, and reflected off the OBCBB.	W/(m <sup>2</sup> μm sr)
$N_{acq}$		Number of scans over which solar diffuser is observed	Unitless
$N_{cal}$	B	Number of frames per scan while observing OBCBB.	Unitless
$N_F$		Along-scan frame number.	Unitless
$N_G$		DNB Gain stage.	Unitless
$N_H$		HAM side index.	Unitless
$N_P$		Along-track pixel.	Unitless
$N_{SDfrm}$	B	The number of frames per scan over which the solar diffuser calibration counts are recorded.	Unitless
$RV(\theta, B)$	d, m	Response Versus Scan function at scan angle $\theta$ for band $B$ .	Unitless
$RVSN_F, N_P, N_H]$		Pre-determined Response Versus Scan for DNB, as a function of $N_F$ , $N_P$ , and $N_H$ .	Unitless
$t$		Time.	Seconds
$t_n$		Time of scan n of an acquisition.	Seconds
$\Delta t_{frame}$	B	Time between frames or samples for detectors.	Seconds
$T_{cav}$		Temperature of cavity contributing to reflection off OBCBB.	K
$T_{det}$		Temperature of FPA detector circuits.	K
$T_{elec}$		Temperature of electronics module.	K
$T_{ham}$		Temperature of HAM.	K
$T_{obc}$		Temperature of OBCBB.	K
$T_{rta}$		Temperature of RTA.	K
$T_{sh}$		Temperature of shield contributing to reflection off OBCBB.	K
$T_{tele}$		Temperature of telescope contributing to reflection off OBCBB.	K
$V_{dcr}$	B, d	Analog DC restore voltage signal with respect to some nominal dark condition.	V

Symbol	Un-notated Indices <sup>2</sup>	Description	Units
$\varepsilon_{obc}(\lambda)$		Spectral emissivity of the OCB	Unitless
$\theta_{inc}$		Incidence angle onto solar diffuser relative to normal.	Radians
$\theta_{obc}$		Scan angle of OCB	Radians
$\theta_{sd}$		Scan angle at solar diffuser.	Radians
$\theta_{ev}$		Scan angle at earth view.	Radians
$\theta_{sun\_earth}$		Solar incidence angle of the sun on the earth.	Radians
$\lambda$		Wavelength.	$\mu\text{m}$
$\overline{\rho_{ev}}(\theta_{ev}, B)$		Band-averaged earth view spectral reflectance.	Unitless
$\rho_{rta}(\lambda)$		Spectral reflectance of RTA.	Unitless
$\tau_{sds}(\phi_h, \phi_v, \lambda)$	d	Spectral transmittance of solar diffuser screen.	Unitless
$\phi_v(t)$		Vertical incidence angle of solar illumination upon SD in laboratory coordinates.	Radians
$\phi_h(t)$		Horizontal incidence angle of solar illumination upon SD in laboratory coordinates.	Radians
$\phi_{inc}$		Azimuthal incidence angle onto solar diffuser relative to x-axis.	Radians
$\Phi_{sun}(\lambda, t)$		Spectral output power of the sun.	W/ $\mu\text{m}$

## 3.4.2.3 Reflective Calibration Core Equations

**Table 3.4.2.3-1. Reflective Calibration Core Equations (Calibration)**

Equations	OAD Eqn
$\overline{L_{ap}}(\theta_{ev}, B) = \frac{\overline{\Delta L_{det}}(\theta_{ev}, B)}{\text{RVS}(\theta_{ev}, B)} = \frac{F \cdot \sum_{j=0}^2 c_j \cdot dn_{ev}^j}{\text{RVS}(\theta_{ev}, B)} = \frac{\sum_{j=0}^2 c'_j \cdot dn_{ev}^j}{\text{RVS}(\theta_{ev}, B)}$	70
$\overline{\rho_{ev}}(\theta_{ev}, B) = \frac{\pi \cdot \sum_{j=0}^2 c'_j dn_{ev}^j}{\text{RVS}(\theta_{ev}, B) \cdot \cos(\theta_{sun\_earth}) \cdot \overline{E_{sun}}(\lambda, d_{se})}$ $= \frac{\sum_{j=0}^2 d_j dn_{ev}^j}{\text{RVS}(\theta_{ev}, B) \cdot \cos(\theta_{sun\_earth})}$	81
$dn_{ev} = DN - \overline{DN}_{sv}$	28
$c'_j = F \cdot c_j$	65
$d_j = \frac{\pi}{\overline{E_{sun}}(\lambda, d_{se})} \cdot c'_j$	76
$C_0(T_{det}, T_{elec}) = \Delta C_0(T_{det}, T_{elec})$	N/A
$C_1(T_{det}, T_{elec}, V_{dcr}) = G \cdot b_1(T_{elec}) \cdot [a_1(T_{det}) + 2a_2(T_{det}) \cdot V_{dcr}] + \Delta C_1(T_{det}, T_{elec})$	N/A
$C_2(T_{det}, T_{elec}, V_{dcr}) = G \cdot [a_1(T_{det}) + 2a_2(T_{det}) \cdot V_{dcr}] \cdot b_2(T_{elec}) + a_2(T_{det}) \cdot b_1(T_{elec})^2 + \Delta C_2(T_{det}, T_{elec})$	N/A

## 3.4.2.4 Emissive Calibration Core Equations

**Table 3.4.2.4-1. Emissive Calibration Core Equations (Calibration)**

Equations	OAD Eqn
$\overline{L}_{ap}(\theta_{ev}, B) = \frac{(1 - RVS(\theta_{ev}, B)) \cdot \left[ \left( \frac{1}{\rho_{rta}(\lambda)} - 1 \right) \cdot \overline{L}(T_{rta}, \lambda) - \frac{\overline{L}(T_{ham}, \lambda)}{\rho_{rta}(\lambda)} \right] + F \cdot \sum_{j=0}^2 c_j(T_{det}, T_{elec}) \cdot dn_{ev}^j}{RVS(\theta_{ev}, B)}$	115
$F = RVS(\theta_{obc}) \cdot \frac{\left\{ \left( 1 - \frac{1}{RVS(\theta_{obc})} \right) \cdot \left[ \left( \frac{1}{\rho_{rta}(\lambda)} - 1 \right) \cdot \overline{L}(T_{rta}(t), \lambda) - \frac{\overline{L}(T_{ham}(t), \lambda)}{\rho_{rta}(\lambda)} \right] \right\} + \overline{\varepsilon}_{obc}(\lambda) \cdot \overline{L}(T_{obc}(t), \lambda) + \overline{L}_{obc\_rfl}(T_{sh}(t), T_{cav}(t), T_{tele}(t), \lambda)}{\sum_{j=0}^2 c_j \cdot \overline{dn}_{obc}(t)^j}$	112
$\overline{L}_{obc\_rfl}(T_{sh}, T_{cav}, T_{tele}, \lambda) = \left[ \begin{array}{l} F_{cav} \cdot (1 - \varepsilon_{obc}(\lambda)) \cdot \overline{L}(T_{cav}, \lambda) + F_{sh} \cdot (1 - \varepsilon_{obc}(\lambda)) \cdot \overline{L}(T_{sh}, \lambda) \\ + F_{tele} \cdot (1 - \varepsilon_{obc}(\lambda)) \cdot \overline{L}(T_{tele}, \lambda) \end{array} \right]$	114
$\overline{dn}_{obc}(t) = \frac{1}{N_{cal}} \sum_{m=0}^{N_{cal}} dn_{obc}(t + m \cdot \Delta t_{frame})$	106
Same $c_j$ values as in Table 2-8	

## 3.4.2.5 Day Night Band Calibration Core Equations

**Table 3.4.2.5-1. Day Night Band Calibration Core Equations (Calibration)**

Equations	OAD Eqn
$\overline{L}_{DNB} = \frac{\sum_{i=0}^2 c_i [agg(N_F, N_P, N_G)] \cdot dn_{DNB}^i}{RVS[N_F, N_P, N_H]}$	119
$dn_{DNB} = DN_{DNB} - DN_{sv\_DNB}[N_F, N_P, N_G]$	120

## 3.4.2.6 Solar Diffuser View Data Processing Core Equations

**Table 3.4.2.6-1. Solar Diffuser View Data Processing Core Equations (Calibration)**

Equations	OAD Eqn
$F = \frac{RVS(\theta_{sd}, B)}{N_{acq}} \cdot \left( \frac{\overline{d}_{se}}{d_{se}} \right)^2 \cdot \sum_{n=0}^{N_{acq}-1} \left( \frac{\cos(\theta_{inc}(t_n)) \cdot \overline{L}_{nsd}(\phi_h(t_n), \phi_v(t_n), B)}{\sum_{j=0}^2 c_j \cdot \overline{dn}_{sd}(t_n)^j} \right)$	68
$\begin{aligned} \overline{L}_{nsd}(\phi_h, \phi_v, B) &= (\tau_{sds}(\phi_h, \phi_v, \lambda) \cdot E_{sun}(\lambda, \overline{d}_{se}) \cdot BRDF(\phi_h, \phi_v, \lambda)) \\ &= \frac{1}{4\pi \cdot \overline{d}_{se}^2} \cdot (\tau_{sds}(\phi_h, \phi_v, \lambda) \cdot \Phi_{sun}(\lambda) \cdot BRDF(\phi_h, \phi_v, \lambda)) \end{aligned}$	67
$\overline{dn}_{sd}(t_n) = \frac{1}{N_{SDfrm}} \sum_{m=0}^{N_{SDfrm}} dn_{sd}(t_n + m \cdot \Delta t_{frame})$	69



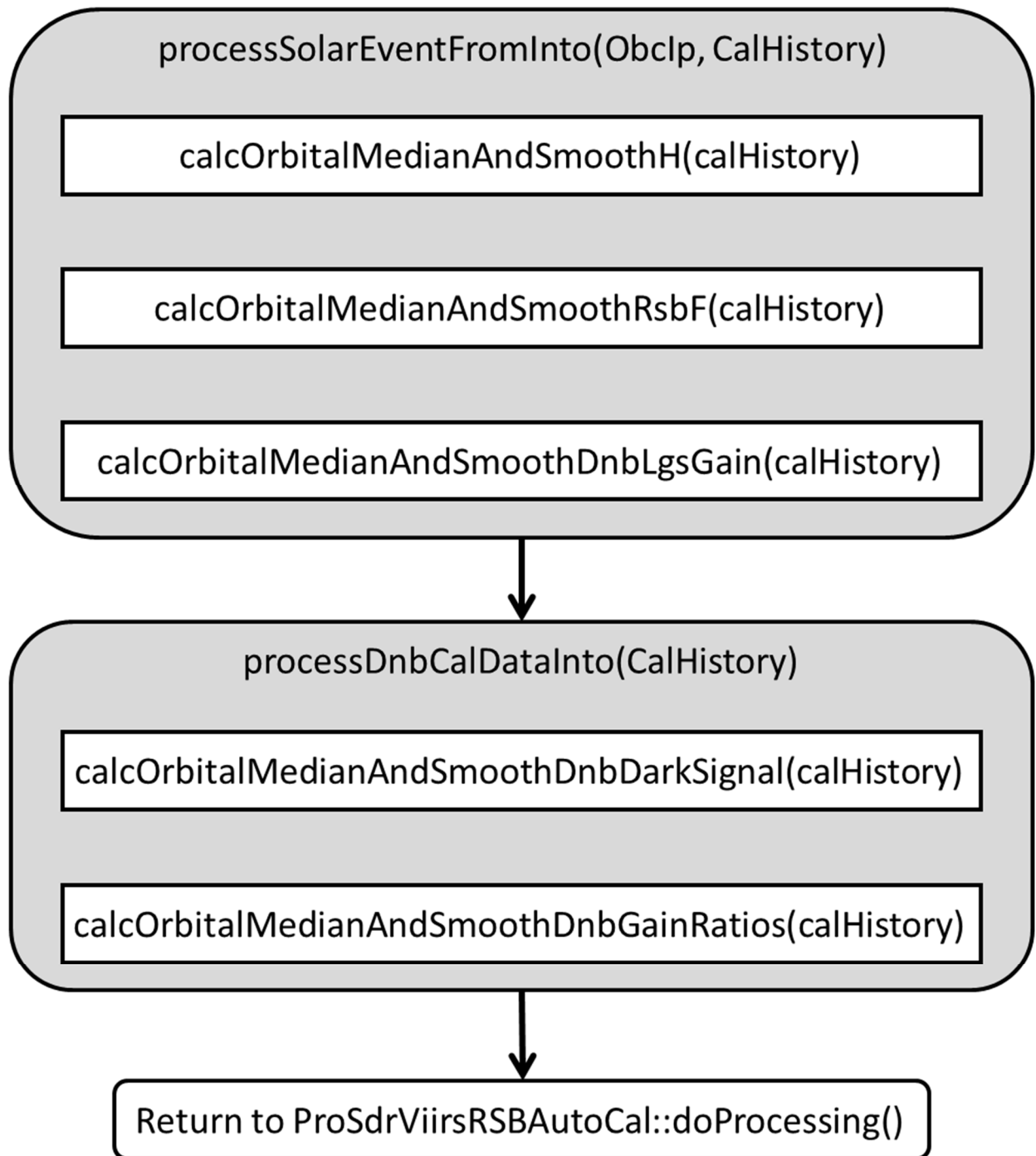
Equations	OAD Eqn
$\phi_v = \arctan\left(\frac{\tan\theta_{inc}}{\sqrt{\tan\phi_{inc}^2 + 1}}\right)$	104
$\phi_h = \arctan\left(\frac{\tan\theta_{inc} \tan\phi_{inc}}{\sqrt{\tan\phi_{inc}^2 + 1}}\right)$	105

### 3.4.2.7 Event Processing

Calibration of the reflective solar bands of the VIIRS instrument happens once per orbit, when the orbit geometry allows it to be illuminated by the sun.

#### 3.4.2.7.1 Solar Event Processing Flow

Figure 3.4.2.7.1-1 shows the SDSM Processing Block diagram.



**Figure 3.4.2.7.1-1. Detailed Solar Event Processing Program Flow**

#### **3.4.2.7.1.1 *calcOrbitalMedianAndSmoothH***

The Solar Diffuser BRDF degradation factors, or “H” factors, are calculated as described in VIIRS Radiometric section 3.3.3.1 for the eight SDSM detectors. The H factors are updated every orbit using measured data from the most recent operation of the SDSM.

#### 3.4.2.7.1.2 *calcOrbitalMedianAndSmoothRsbF*

The RSB calibration scale factors, or F factors, are calculated as described in VIIRS Radiometric Calibration section 3.3.3. The F factors are computed every orbit shortly after the solar calibration data are acquired.

#### 3.4.2.7.1.3 *calcOrbitalMedianAndSmoothDnbLgsGain*

The DNB LGS gain is calculated as described in VIIRS Radiometric Calibration section 3.3.5.3 for each aggregation zone, detector, and mirror side. The gains are computed every orbit in this algorithm shortly after the solar calibration data is acquired.

#### 3.4.2.7.1.4 *calcOrbitalMedianAndSmoothDnbDarkSignal*

The DNB dark signal offsets are determined from Earth View (EV) data acquired during new moon periods when the instrument is viewing dark, uniform ocean scenes and from calibration sector data acquired every scan (Solar Diffuser (SD), On-Board Calibrator Blackbody (OBC BB), and Space View (SV)). The EV dark scene data are used to generate offsets for every detector, every sample across the EV scan, both Half Angle Mirror (HAM) sides and all three DNB gain stages.

#### 3.4.2.7.1.5 *calcOrbitalMedianAndSmoothDnbGainRatios*

DNB LGS gain ratios are calculated from the calibration sector data acquired over the terminator region. Pairs of unsaturated signals for successive gain states, e.g., MGS and LGS, MGS and HGA, or MGS and HGB, are identified and captured. For pairs of dark signals that pass through both filters, gain ratios are calculated as follows:

$$\frac{c_{MGS}}{c_{LGS}} = \frac{dn_{LGS}}{dn_{MGS}}$$

$$\frac{c_{HGA}}{c_{MGS}} = \frac{1}{2} \frac{dn_{MGS}}{dn_{HGA}}$$

$$\frac{c_{HGB}}{c_{MGS}} = \frac{dn_{MGS}}{dn_{HGB}}$$

where:

$c_{MGS}$  = MGS gain for 13 bit EV data

$c_{LGS}$  = LGS gain for 13 bit EV data

$c_{HGA}$  = HGA gain for 14 bit EV data

$c_{HGB}$  = HGB gain for 14 bit EV data

$dn_{LGS}$  = LGS offset corrected cal sector counts at 14 bits

$dn_{MGS}$  = MGS offset corrected cal sector counts at 14 bits

$dn_{HGA}$  = HGA offset corrected cal sector counts at 14 bits

$dn_{HGB}$  = HGB offset corrected cal sector counts at 14 bits

Once the gain ratios have been calculated for a given orbit for each calibration sector, detector, sequence order, and mirror side, the values for different calibration sectors are pooled together

---

into a common set, depending upon the value of a flag in a LUT. Any subset of the three calibration sectors can be selected for use in dark signal determination. The median of the pooled set is calculated as the representative value of the gain ratio for the given orbit.

### 3.4.3 Graceful Degradation

The VIIRS Reflective Solar Band Automated Calibration process contains no graceful degradation.

#### 3.4.3.1 Graceful Degradation Inputs

None.

#### 3.4.3.2 Graceful Degradation Processing

None.

#### 3.4.3.3 Graceful Degradation Outputs

None.

### 3.4.4 Exception Handling

No exception handling is performed in the VIIRS SDR Solar Diffuser process.

### 3.4.5 Data Quality Monitoring

No data quality monitoring is performed.

### 3.4.6 Computational Precision Requirements

The VIIRS Reflective Solar Band Automated Calibration process uses double precision values in order to ensure the required accuracy.

### 3.4.7 Algorithm Support Considerations

The DMS and INF must be running before the algorithm is executed.

### 3.4.8 Assumptions and Limitations

The Reflective Solar Band Automated Calibration Processing Unit needs the inputs identified in Section 3.4.1.1 before the execution of the code.

## 3.5 VIIRS SDR Bright Pixel Description

Scattered light can contaminate pixels in the vicinity of bright objects in VIIRS scenes. Also, bright objects can lose light that is scattered to darker areas. Although the fraction of scattered light is low, for a sufficiently bright source near a dim nearby pixel, the scattered contribution can dominate the radiometric uncertainty. This condition is easily understood by considering the ocean's color and VIIRS ability to determine that color with a bright cloud in the nearby scene. Several EDRs (ocean color, land albedo, and sea-surface temperature) have exclusion conditions in the 474-00448-04-06\_JPSS-SRSPF-Vol-IV-Part-6\_B\_VIIRS-RDR-SDR.xml around bright pixels, and call for a flag to indicate when scattering is above an acceptable level.

### 3.5.1 Interfaces

#### 3.5.1.1 Inputs

VIIRS Bright Pixel Inputs are listed in 474-00448-01-06\_JPSS-SRS-Vol-I-Part-06, Table 3-1 (rows 3 & 11). Inputs are further defined below:

**Table 3.5.1.1-1. VIIRS SDR Bright Pixel Inputs**

Input	Description	Reference Document
VIIRS Mod Band SDR	VIIRS M1-M16 750m (Moderate) SDRs	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Bright Pixel PSF MOD LUT	The VIIRS Bright Pixel PSF MOD LUT file contains the Point Spread Function (PSF) parameters for use in determining Bright Pixels	474-00448-02-06_JPSS-DD-Vol-II-Part-06
VIIRS Bright Pixel Sub Radiance LUT	The VIIRS Bright Pixel Sub Radiance LUT file contains substitute radiance values used for saturated pixels.	474-00448-02-06_JPSS-DD-Vol-II-Part-06
Bright Pixel Threshold LUT	The VIIRS Bright Pixel Flag Threshold LUT file contains bit patterns that match corresponding percents of scattered light. These thresholds are used to determine the output 4-bit state flag that is the output of the Bright Pixel Algorithm.	474-00448-02-06_JPSS-DD-Vol-II-Part-06

#### 3.5.1.2 Outputs

The output of this algorithm will be a 4-bit flag for each pixel in each band that characterizes the level of contamination of signal by scattered light.

The percentage of scattered light is defined to be:

$$\% \text{ scattered light} = \left| \frac{S_{scat}}{S_{meas}} \right| * 100$$

4-bit state flag:

0000	0.0 - 0.01% scattered light
0001	0.01 - 0.02% scattered light
0010	0.02 - 0.05% scattered light
0011	0.05 – 0.1% scattered light
0100	0.1 – 0.2% scattered light
0101	0.2 – 0.5% scattered light
0110	0.5 - 1.0% scattered light
0111	1.0 - 2.0% scattered light
1000	2.0 – 5.0% scattered light
1001	5 – 10% scattered light
1010	>10% scattered light
1111	calculation unreliable

VIIRS Bright Pixel Output is listed in 474-00448-01-06\_JPSS-SRS-Vol-I-Part-06, Table 3-1 (row 27).

**Table 3.5.1.2-1. VIIRS Bright Pixel Output**

Output	Description	Reference Document
VIIRS Bright Pixel IP	Flag containing the Bright Pixel exclusion condition for the OCC and Albedo EDRs. The output for the IP is a value for each pixel in each band that characterizes the level of contamination of signal by scattered light	474-00448-02-06_JPSS-DD-Vol-II-Part-06

### 3.5.2 Algorithm Processing

#### 3.5.2.1 Estimating % Scattered Light

Scattered light distribution is a property of the telescope & focal plane and is referred to as the near-field scattering. Bright Pixel Algorithm models the scattering with a point-spread function that is composed of a delta function with small epsilon tails. The measured scene is the true scene convolved with the PSF.

Definitions:

$$S_{true} = \text{true scene}$$

$$S_{meas} = \text{measured scene including scattered light}$$

$$P_{true} = \text{true PSF}$$

$$P_{meas} = \text{measured PSF}$$

$$\delta = \text{Dirac Delta Function}$$

$$\delta(i, j) = \begin{cases} 1 & \text{if } i = i_{center} \text{ \& } j = j_{center} \\ 0 & \text{otherwise} \end{cases}$$

$$i = \text{in-track sample index}$$

$$j = \text{in-scan sample index}$$

$$S_{meas} = P_{true} * S_{true}$$

The PSF here is a full optical PSF, including all the effects of Near-Field Scattering (NFS). This is sometimes also referred to as a Point Source Transmittance (PST), but it will be referred to as PSF here. The effect of NFS can be considered as a perturbation,  $\epsilon$ , with respect to the Dirac Delta function

$$P_{true} = \delta + \epsilon_{true}$$

$$P_{meas} = \delta + \epsilon_{meas}$$

The following normalization conditions apply to both the measured and the true PSF.

$$\sum_i \sum_j P(i, j) = 1$$

$$\sum_i \sum_j \epsilon(i, j) = 0$$

The true scattering in the scene is the difference between the measured scene and the true scene.

$$S_{scat} = S_{meas} - S_{true} = P_{true} * S_{true} - \delta * S_{true}$$

$$= (P_{true} - \delta) * S_{true}$$

$$= \epsilon_{true} * S_{true}$$

If the measured scene is convolved again with the PSF, it is found:

$$\begin{aligned} S_{double\ conv} &= P_{meas} * S_{meas} \\ &= (\delta + \varepsilon_{meas}) * (\delta + \varepsilon_{true}) * S_{true} \\ &= (\delta * \delta + \varepsilon_{meas} * \varepsilon_{true} + \varepsilon_{meas} + \varepsilon_{true}) * S_{true} \end{aligned}$$

Subtracting the measured scene from this gives what will be referred to as  $S'_{scat}$ .

$$\begin{aligned} S'_{scat} &= S_{double\ conv} - S_{meas} = P_{meas} * S_{meas} - \delta * S_{meas} \\ &= (\delta + \varepsilon_{meas} - \delta) * S_{meas} \\ &= \varepsilon_{meas} * S_{meas} \\ &= \varepsilon_{meas} * (\delta + \varepsilon_{true}) * S_{true} \end{aligned}$$

Then it can be written:

$$\begin{aligned} S'_{scat} - S_{scat} &= (\varepsilon_{meas} + \varepsilon_{meas} * \varepsilon_{true} - \varepsilon_{true}) * S_{true} \\ \text{or} \\ S'_{scat} &= S_{scat} + (\varepsilon_{meas} - \varepsilon_{true} + \varepsilon_{meas} * \varepsilon_{true}) * S \end{aligned}$$

which reduces to  $S'_{scat} = S_{scat}$  when  $\varepsilon_{true} \cong \varepsilon_{meas}$  and  $\varepsilon_{true} * \varepsilon_{meas} * S \cong 0$ .

As long as the measured NFS is a good estimate of the true NFS, and NSF is a small contribution to the whole PSF, then the difference between the doubly convolved scene and the measured scene is a good estimate of the scattering. Our estimate of scattered light then is:

$$\% \text{ scattered light} = \frac{S_{scat}}{S_{meas}} * 100 \cong \frac{S'_{scat}}{S_{meas}} * 100 \cong \frac{\varepsilon_{meas} * S_{meas}}{S_{meas}} * 100$$

In addition to the above requirements that the measured scattering distribution be close to the real one and much smaller than one, there is another requirement here, that  $S_{meas} \gg S'_{scat}$ . Very close to a bright object, where the % scattered light is close to or greater than one, the exact value will become very uncertain. However, there is no uncertainty that the scattering is large, but the only thing that is uncertain is how large the scattering is going to be. Since it has been chosen that highest threshold of 10% for the flags, anything above this is flagged as >10% scattered light, so it will not matter whether it is, say, 20% or 40%. Either way, it is much too large an error to provide a meaningful EDR.

### $\varepsilon_{meas}$ as an estimate of $\varepsilon_{true}$

SBRS measures the LSF and Near-Field scattering. Until such measurements become available for the flight unit, the in-track and cross-track NFS contribution to the PSF will be based on a fit to a Harvey-Shack (H-S) BRDF model. H-S BRDF is translated into  $\varepsilon_{meas}$ , using the angular separations in the BRDF to describe the scattering to adjacent samples in the in-track and in-scan direction.

$$\text{PSF} = \delta + \varepsilon_{\text{meas}}$$

The true scattering function,  $\varepsilon_{\text{true}}$ , is expected to differ from this, especially over the life of VIIRS as optical surfaces degrade with time, and SBRS has models to estimate this impact. So that our flags are conservatively pessimistic, end-of-life (EOL) estimates are to be used over the entire mission.

### 3.5.2.2 “Calculation Unreliable” Flag

The tails of the scattering distribution are notoriously difficult to measure since they are several orders of magnitude lower than the peak. The VIIRS near-field response (NFR) test FP-14 uses bright sources that saturate the central pixels to measure the tails and therefore it is expected to know  $\varepsilon_{\text{meas}}$  over several orders of magnitude. There will be some point, however, when the measured tails become unreliable. Computing the scattering fraction becomes problematic when both  $S_{\text{meas}}$  and  $S'_{\text{scat}}$  become very small, since the uncertainty of both is inversely related to the value.

A reasonable approach is to set minimum thresholds for each. The flag is then set if both  $S_{\text{meas}}$  and  $S'_{\text{scat}}$  are below their associated thresholds. For  $S_{\text{meas}}$ , a dim pixel threshold  $S_{\text{lo\_thresh}}$  is defined for each band. A reasonable choice for  $S_{\text{lo\_thresh}}$  is Lmin or, for the emissive bands, is the black body radiance associated with Tmin.

For  $S'_{\text{scat}}$  a threshold,  $S_{\text{scat\_thresh}}$ , will be based on test data. Determine the minimum  $\varepsilon_{\text{meas}}$  that is reliable and flag possible conditions where a reference bright target could scatter significant light beyond the measured tails. The bright target, as defined by the spec is at Lmax and has an angular extent of 12 by 12 milliradians per the System Spec. If  $\theta$  is the angle which defines the limit of the measurement of the NFS, then  $S_{\text{scat\_thresh}} = S'_{\text{scat}}(\theta)$  where  $S'_{\text{scat}}(\theta)$  is computed for the bright target.

### 3.5.2.3 Creation of Non-Saturated Scene

Determining the % scattered light using the measured scene and a PSF has been considered; however, there is a difference between the scene needed to do this computation and a VIIRS SDR. The SDR is not a continuous field of radiance, which is what is necessary to do the convolution described above. Therefore, there are several features of the VIIRS SDR which need to be addressed to produce a continuous scene from the SDR.

- **Saturated radiance**

Values are assigned from a proxy band that has highly correlated radiances and substitute scaled radiances of the proxy band in place of the saturated data in cases where the pixel is saturated. Which proxy bands to use and the scale factors are controlled by a changeable LUT.



If no suitable proxy band is found, or if the radiance in the proxy band is saturated or missing, then default radiance based on maximum reflectance or brightness temperature are used when available. This would be higher than Tmax or Lmax values and lead to a conservative upper estimate of the scattering.

- **In-scan angular resolution and Aggregation**

The PSF is a function of uniformly spaced pixels on the focal plane. Because of the size of the scattering PSF it is faster to convolve using an FFT. Computationally an FFT convolution requires a regular sample interval. Because the in-scan angular resolution changes across an aggregation zone boundary, each aggregation zone has a different angular resolution in-scan. To overcome this problem the aggregated pixels are “de-aggregated” by repeating them by the number of samples that were used in the original aggregation. A guard band would be added that would avoid aliasing of the FFT.

- **Bowtie and Edge of scan effects**

Re-pixelize is needed between the scans because of the “bowtie” overlap with adjacent scans. IDPS uses a common adjacency implementation to correct bowtie and edge of scan effects. This implement is not the same as is used in the science grade code which causes difference the outputs of the science algorithm to the science and operational implementation.

- **Missing radiance**

If the gap in radiance data is small, such as a single pixel, or a row of pixels, then the pixels are estimated by interpolating the nearest neighbors. If the gap is large, then it is filled in with proxy bands.

- **Scan & Granule limits**

The PSF is created to be as wide as or wider than a scan in the in-track direction. Reliable convolution depends on having guard regions around the scene that are about the size of the PSF. Granules are processed so that the granule before and the granule after is always available, so that scans adjacent to the first and last scan of the granule are available.

- **Scene edges**

In the usual processing mode, there is always a granule before and after the current granule. In the case where the process is just starting up, the IDPS algorithm should create a guard region by reflecting the first or last granule in the processing chain so that the flags can be calculated. The alternative is to set the “calculation unreliable” flag until the 2<sup>nd</sup> or 3<sup>rd</sup> scan. This is not part of the Science Grade Code. The IDPS Ops code uses the Common adjacency solution to pad the data with the previous and next scans providing all needed data for calculation.

#### 3.5.2.4 Common Adjacency Use In Bright Pixel

On Board Pixel trim in the SDR data must be replaced in order to provide for continuous data in on board pixel trim regions of each scan before convolution with the point spread function for calculation of the stray light percentages can be done. This replacement was described in the TM as being an area that must be modified by the IDPS operational baseline to use a modified version of the common IDPS operation method of on board bowtie deleted pixel replacement.

---

The provided implementation uses a spline average replacement as done with the bad data replacement.

The first step in the replacement is to find the nearest neighbor for the bowtie deleted pixels in the previous or next scan as needed. The nearest neighbor once identified is then copied into the bowtie area, along with the in-track neighbors to the identified replacement pixel filling all missing bowtie pixels in the in-track direction in current scan being processed. This process is repeated for each cross-track (column) for all onboard bowtie deleted pixels. This is then continued for all scans of data being processed.

#### 3.5.2.5 Data Convolution

In the TM provided, along with explaining the needed changes for the on board pixel trim replacement, it also described a new approach for doing the point spread function convolution to replace the existing science implementation. The new approach is meant to remove discontinuities that can be produced during the convolution of the data. The science implementation takes the SDR data that has been updated to replace bad pixels, saturated pixels, and bowtie pixels and pads the data with the points spread function data and convolutes it with the PSF (point spread function) data. This is implemented in the science code as a by band calculation done once for each band.

IDPS operational implementation follows the method outlined in the TM by instead convoluting each scan separately in a 5-scan matrix. The 5-scan matrixes are made of the current scan as scan 3 of the 5-scan matrix. Scans 1 and 2 are made up of the previous two scans, and scans 4 and 5 are made using the next two scans. After the convolution is completed, only the current scan (scan 3 in the matrix) is maintained for use after the convolution.

Further differences were needed to implement this approach in the IDPS operational baseline for efficiency reasons. The code was modified to process the bowtie replacement on each matrix instead of the whole granule at once. The modification of the data from aggregated to unaggregated VIIRS space were also modified and moved to be done on the matrix of data to prevent duplication of work. Only the current scan is aggregated at the end to make the code more efficient.

#### 3.5.3 Graceful Degradation

The VIIRS Bright Pixel process contains no graceful degradation.

##### 3.5.3.1 Graceful Degradation Inputs

None.

##### 3.5.3.2 Graceful Degradation Processing

None.

##### 3.5.3.3 Graceful Degradation Outputs

None.

#### 3.5.4 Exception Handling

No exception handling is performed in the VIIRS Bright Pixel process.

### 3.5.5 Data Quality Monitoring

No data quality monitoring is performed.

### 3.5.6 Computational Precision Requirements

The VIIRS Bright Pixel process uses double precision values in order to ensure the required accuracy.

### 3.5.7 Algorithm Support Considerations

The DMS and INF must be running before the algorithm is executed.

### 3.5.8 Assumptions and Limitations

Bright Pixel implementation is currently limited to Moderate band data processing. Image band support is available with minor updates to the code.

## 4 GLOSSARY/ACRONYM LIST

### 4.1 Glossary

Below is a glossary of terms most applicable for this OAD.

Term	Description
Algorithm	A formula or set of steps for solving a particular problem. Algorithms can be expressed in any language, from natural languages like English to mathematical expressions to programming languages like FORTRAN. On JPSS, an algorithm consists of: <ol style="list-style-type: none"> <li>1. A theoretical description (i.e., science/mathematical basis)</li> <li>2. A computer implementation description (i.e., method of solution)</li> <li>3. A computer implementation (i.e., code)</li> </ol>
Algorithm Engineering Review Board (AERB)	Interdisciplinary board of scientific and engineering personnel responsible for the approval and disposition of algorithm acceptance, verification, development and testing transitions. Chaired by the Data Process Algorithm Lead, members include representatives from STAR, DPMS, IDPS, and Raytheon..
Algorithm Verification	Science-grade software delivered by an algorithm provider is verified for compliance with data quality and timeliness requirements by Algorithm Team science personnel. This activity is nominally performed at the GRAVITE facility. Delivered code is executed on compatible GRAVITE computing platforms. Minor hosting modifications may be made to allow code execution. Optionally, verification may be performed at the Algorithm Provider's facility if warranted due to technical, schedule or cost considerations.
Ancillary Data	Any data which is not produced by the JPSSSystem, but which is acquired from external providers and used by the JPSSsystem in the production of JPSSdata products.
Auxiliary Data	Auxiliary Data is defined as data, other than data included in the sensor application packets, which is produced internally by the JPSSsystem, and used to produce the JPSSdeliverable data products.
EDR Algorithm	Scientific description and corresponding software and test data necessary to produce one or more environmental data records. The scientific computational basis for the production of each data record is described in an OAD. At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Environmental Data Record (EDR)	<p><i>[IORD Definition]</i></p> <p>Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to geophysical parameters (including ancillary parameters, e.g., cloud clear radiation, etc.).</p> <p><i>[Supplementary Definition]</i></p> <p>An Environmental Data Record (EDR) represents the state of the environment, and the related information needed to access and understand the record. Specifically, it is a set of related data items that describe one or more related estimated environmental parameters over a limited time-space range. The parameters are located by time and Earth coordinates. EDRs may have been resampled if they are created from multiple data sources with different sampling patterns. An EDR is created from one or more JPSSDRs or EDRs, plus ancillary environmental data provided by others. EDR metadata contains references to its processing history, spatial and temporal coverage, and quality.</p>
Model Validation	The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model.
Model Verification	The process of determining that a model implementation accurately represents the developer's conceptual description and specifications.
Operational Code	Verified science-grade software, delivered by an algorithm provider and verified by GRAVITE, is developed into operational-grade code by the IDPS IPT.
Operational-Grade Software	Code that produces data records compliant with the System Specification requirements for data quality and IDPS timeliness and operational infrastructure. The software is modular relative to the IDPS infrastructure and compliant with IDPS application programming interfaces (APIs) as specified for TDR/SDR or EDR code.

Term	Description
Raw Data Record (RDR)	<p><i>[IORD Definition]</i></p> <p>Full resolution digital sensor data, time referenced and earth located, with absolute radiometric and geometric calibration coefficients appended, but not applied, to the data. Aggregates (sums or weighted averages) of detector samples are considered to be full resolution data if the aggregation is normally performed to meet resolution and other requirements. Sensor data shall be unprocessed with the following exceptions: time delay and integration (TDI), detector array non-uniformity correction (i.e., offset and responsivity equalization), and data compression are allowed. Lossy data compression is allowed only if the total measurement error is dominated by error sources other than the data compression algorithm. All calibration data will be retained and communicated to the ground without lossy compression.</p> <p><i>[Supplementary Definition]</i></p> <p>A Raw Data Record (RDR) is a logical grouping of raw data output by a sensor, and related information needed to process the record into an SDR or TDR. Specifically, it is a set of unmodified raw data (mission and housekeeping) produced by a sensor suite, one sensor, or a reasonable subset of a sensor (e.g., channel or channel group), over a specified, limited time range. Along with the sensor data, the RDR includes auxiliary data from other portions of JPSS (space or ground) needed to recreate the sensor measurement, to correct the measurement for known distortions, and to locate the measurement in time and space, through subsequent processing. Metadata is associated with the sensor and auxiliary data to permit its effective use.</p>
Retrieval Algorithm	A science-based algorithm used to 'retrieve' a set of environmental/geophysical parameters (EDR) from calibrated and geolocated sensor data (SDR). Synonym for EDR processing.
Science Algorithm	The theoretical description and a corresponding software implementation needed to produce an NPP/JPSS data product (TDR, SDR or EDR). The former is described in an OAD. The latter is typically developed for a research setting and characterized as "science-grade".
Science Algorithm Provider	Organization responsible for development and/or delivery of TDR/SDR or EDR algorithms associated with a given sensor.
Science-Grade Software	Code that produces data records in accordance with the science algorithm data quality requirements. This code, typically, has no software requirements for implementation language, targeted operating system, modularity, input and output data format or any other design discipline or assumed infrastructure.
SDR/TDR Algorithm	Scientific description and corresponding software and test data necessary to produce a Temperature Data Record and/or Sensor Data Record given a sensor's Raw Data Record. The scientific computational basis for the production of each data record is described in an Operational Algorithm Document (OAD). At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Sensor Data Record (SDR)	<p><i>[IORD Definition]</i></p> <p>Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to calibrated brightness temperatures with associated ephemeris data. The existence of the SDRs provides reversible data tracking back from the EDRs to the Raw data.</p> <p><i>[Supplementary Definition]</i></p> <p>A Sensor Data Record (SDR) is the recreated input to a sensor, and the related information needed to access and understand the record. Specifically, it is a set of incident flux estimates made by a sensor, over a limited time interval, with annotations that permit its effective use. The environmental flux estimates at the sensor aperture are corrected for sensor effects. The estimates are reported in physically meaningful units, usually in terms of an angular or spatial and temporal distribution at the sensor location, as a function of spectrum, polarization, or delay, and always at full resolution. When meaningful, the flux is also associated with the point on the Earth geoid from which it apparently originated. Also, when meaningful, the sensor flux is converted to an equivalent top-of-atmosphere (TOA) brightness. . The associated metadata includes a record of the processing and sources from which the SDR was created, and other information needed to understand the data.</p>

Term	Description
Temperature Data Record (TDR)	<p><i>[IORD Definition]</i></p> <p>Temperature Data Records (TDRs) are geolocated, antenna temperatures with all relevant calibration data counts and ephemeris data to revert from T-sub-a into counts.</p> <p><i>[Supplementary Definition]</i></p> <p>A Temperature Data Record (TDR) is the brightness temperature value measured by a microwave sensor, and the related information needed to access and understand the record. Specifically, it is a set of the corrected radiometric measurements made by an imaging microwave sensor, over a limited time range, with annotation that permits its effective use. A TDR is a partially-processed variant of an SDR. Instead of reporting the estimated microwave flux from a specified direction, it reports the observed antenna brightness temperature in that direction.</p>

## 4.2 Acronyms

Below is a list of acronyms most applicable for this OAD.

Acronym	Description
AM&S	Algorithms, Models & Simulations
API	Application Programming Interfaces
ARP	Application Related Product
DMS	Data Management Subsystem
DQTT	Data Quality Test Table
E&A	Ephemeris and Attitude
EV	Earth View
FPA	Focal Plane Array
HAM	Half Angle Mirror
H-S	Harvey-Shack
IEO	Instrument Engineering Order
IET	IDPS Epoch Time
IMG	Imagery
INF	Infrastructure
ING	Ingest
IP	Intermediate Product
IPO	Input Processing Output
LOS	Loss of Signal
LUT	Look-Up Table
MBN	Max Band Number
MOD	Moderate
NCSA	National Center for Supercomputing Applications
NFR	Near-Field Response
NFS	Near-Field Scattering
OBC	On-board Calibrator
OBCBB	On-board Calibrator BlackBody
PO	Product Order
PSF	Point Spread Function
PST	Point Source Transmittance
QF	Quality Flag
RTA	Rotating Telescope Assembly
SDR	Sensor Data Record
SDSM	Solar Diffuser Stability Monitor
SI	International System of Units
TBD	To Be Determined
TBR	To Be Resolved
TOA	Top of the Atmosphere
USES	Universal Source Encoder for Space