Effective Date: March 13, 2017
Revision D

GSFC JPSS CMO
May 1, 2017
Released

Joint Polar Satellite System (JPSS) Ground Project
Code 474
474-00093

Joint Polar Satellite System (JPSS) Operational Algorithm Description (OAD) Document for VIIRS Ground Track Mercator (GTM) Imagery Environmental Data Record (EDR) Software

For Public Release

The information provided herein does not contain technical data as defined in the International Traffic in Arms Regulations (ITAR) 22 CFC 120.10. This document has been approved For Public Release to the NOAA Comprehensive Large Array-data Stewardship System (CLASS).

Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics and
Space Administration

Joint Polar Satellite System (JPSS) Operational Algorithm Description (OAD) Document for VIIRS Ground Track Mercator (GTM) Imagery Environmental Data Record (EDR) Software JPSS Electronic Signature Page

Prepared By:

Ray Godin
JPSS Data Products and Algorithms EDR Lead
(Electronic Approvals available online at (https://jpssmis.gsfc.nasa.gov/mainmenu_dsp.cfm)

Approved By:

Gilberto Vicente
JPSS Ground Project Algorithm Integration Team (AIT) Manager
(Electronic Approvals available online at (https://jpssmis.gsfc.nasa.gov/mainmenu_dsp.cfm)

Goddard Space Flight Center
Greenbelt, Maryland

Preface

This document is under JPSS Ground Algorithm ERB 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

Revision	Effective Date	Description of Changes (Reference the CCR & CCB/ERB Approve Date)
Original	06/03/2011	This version incorporates **474-CCR-11-0108** which converts D42815, Operational Algorithm Description Document for VIIRS GTM Imagery EDR Software, Rev A dated 03/13/2009 to a JPSS document, Rev -. This was approved by the JPSS Ground Algorithm ERB on June 3, 2011.
Revision A	01/18/2012	**474-CCR-11-0280:** This version baselines 474-00093, Joint Polar Satellite System (JPSS) Operational Algorithm Description (OAD) Document for VIIRS Ground Track Mercator (GTM) Imagery Environmental Data Record (EDR) Software, for the Mx 6 IDPS release. This CCR was approved by the JPSS Algorithm ERB on January 18, 2012.
Revision B	05/14/2013	**474-CCR-13-0948:** This version authorizes 474-00093, JPSS OAD Document for VIIRS GTM EDR Software, for the Mx 7.0 IDPS release. Includes Raytheon PCR032720; 474-CCR-13-0916/ECR-ALG-0037: Update applicable OAD filenames/template/Rev/etc. for Mx7 Release.
Revision C	05/13/2015	**474-CCR-15-2428:** This version authorizes 474-00093, JPSS OAD Document for VIIRS GTM EDR Software, for the Mx 8.9 IDPS release. Includes Raytheon 46821; PRO: OAD: 474-CCR-15-2345: VIIRS Radiance and Reflectance/Brightness Temperature Upper Bounds & Quality Flagging Are Inconsistent - DR 7294 GTM EDR OAD, in Tables 1, 23, 24 and 29.
Revision D	03/13/2017	**474-CCR-17-3243 (ECR-CGS-0734):** This version authorizes 474-00093, JPSS OAD Document for VIIRS GTM EDR Software, for the Block 2.0 IDPS release. Includes Raytheon PCR045678**:** Block 2.0: PRO: OAD: CCR: 474-CCR-15-2444: General OAD Clean-up CCR/PCR, affects all 35/37 OADs. All sections and tables may be affected.

NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR THE VIIRS Ground Track Mercator (GTM) IMAGERY EDR Software

SDRL No. S141
SYSTEM SPECIFICATION SS22-0096

RAYTHEON COMPANY
INTELLIGENCE AND INFORMATION SYSTEMS (IIS)
NPOESS PROGRAM
OMAHA, NEBRASKA

IAW DFAR 252.227-7036, Raytheon hereby declares that, to the best of its knowledge and belief, the technical data delivered under Subcontract No. 7600002744 is complete, accurate, and complies with all requirements of the Subcontract.

TITLE: NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS) OPERATIONAL ALGORITHM DESCRIPTION DOCUMENT FOR THE VIIRS GTM IMAGERY EDR SOFTWARE

APPROVAL SIGNATURES:

____

Stephen E. Ellefson Date
ING/PRO Lead

____

Gabriela A. Ostler Date
Mission Assurance and Enterprise Effectiveness (MAEE)

Northrop Grumman Space & Mission Systems Corp.
Space Technology
One Space Park
Redondo Beach, CA 90278

Engineering & Manufacturing Development (EMD) Phase
Acquisition & Operations Contract

CAGE NO. 11982

Operational Algorithm Description
VIIRS GTM Imagery EDR Software

Document Date: Sep 26, 2011

Document Number: D42815
Revision: B6

PREPARED BY:

Stephanie Weiss *AM&S Imagery EDR Lead*	*Date*	Paul D. Siebels *IDPS PRO SW Manager*	*Date*

ELECTRONIC APPROVAL SIGNATURES:

Roy Tsugawa *A&DP Lead & ACCB Chair*	*Date*	Stephen E. Ellefson *IDPS Processing SI Lead*	*Date*
Bob Hughes *A&DP Deputy & ARB Chair*	*Date*		

Prepared by
Northrop Grumman Space Technology
One Space Park
Redondo Beach, CA 90278

Prepared for
Department of the Air Force
NPOESS Integrated Program Office
C/O SMC/CIK
2420 Vela Way, Suite 1467-A8
Los Angeles AFB, CA 90245-4659

Under
Contract No. F04701-02-C-0502

This document has been identified per the NPOESS Common Data Format Control Book – External Volume 5 Metadata, D34862-05, Appendix B as a document to be provided to the NOAA Comprehensive Large Array-data Stewardship System (CLASS) via the delivery of NPOESS Document Release Packages to CLASS.

Northrop Grumman Space & Mission Systems Corp.
Space Technology
One Space Park
Redondo Beach, CA 90278

Revision/Change Record

Document Number **D42815**

Revision	Document Date	Revision/Change Description	Pages Affected
---	6-30-04	Initial Release.	All
A1		22Nov05 – Raytheon-Omaha corrected for SOM versus GTM philosophy, inserted new logo, removed export markings, and worked on delta Sci2Ops code conversion process. 01Mar06 – Did edit to change title, convert document from I-Channel specific to GTM Imagery, and add M-Channel algorithm and configuration information. 14Mar06 – Did edit to correct formatting after insertion of Table 5.2-1 M-Band Unit Test dated 3/08/06 (4 pages), update to Table of Contents and List of Tables, plus copyright date on coversheet. 22Mar06 – Updated short names and GTM description from Peer Review actions. 29Mar06 – Updated the List of Figures to capture M-Channel Basic Processing Flow diagram, insert a TBD/TBR page, and make other minor edits. 22May06 – Added reference to EDRPR version. Updated reference to number of products produced for M-Band and fixed configuration guide description for B1.4. 25Oct06 – Updated OAD to include the four additional output bands. 04Jan07 – Updated OAD from GTM NCC Imagery Peer Review to convert NCC Imagery to a GTM Imagery algorithm per ECR-480.	All
A2	4-10-07	Updated wording of Detailed Algorithm description, updated to new logo.	All
A3	4-27-07	GTM FO dead detector replacement alg.	All
A4	5-3-07	Changed Table 2.2.2 to show latitude and longitude in degrees.	All
A5	5-18-07	Included updates as per NP-EMD.2006.510.0079-Rev A.	All
A6	10-29-07	Updated for code changes made under ECR-547.	All
A7	11-16-07	Changes made to OAD based on comments from Stephanie Weiss.	All
A8	11-21-07	Delivered to NGST.	All
A9	9-15-08	Reformatted to conform to the D41851 template. Updated data quality notification sections. New cover sheet, update references, acronym list, prepare for peer review. Delivered to NGST. Accept all changes after delivery.	All
A10	02-18-09	Updated with SDRL comments for TIM.	All
A	3-13-09	Incorporated TIM comments and final preparations for ARB/ACCB.	All
B1	12-01-09	Resolved RFA Nos. 310 & 311 and updated subcontract number.	Title pg
B2	06-01-10	Updated for TIM	All

Northrop Grumman Space & Mission Systems Corp.
Space Technology
One Space Park
Redondo Beach, CA 90278

NORTHROP GRUMMAN

Raytheon

Revision/Change Record

Document Number **D42815**

Revision	Document Date	Revision/Change Description	Pages Affected
B3	07-08-10	Updated for OAD Giver UID 35077	All
B4	08-23-10	Updated to reflect changes after GTM rework	All
B5	10-14-10	Updated due to document convergence	All
B6	09-26-11	Updated OAD for PCR026647.	All

Table of Contents

List of Figures

List of Tables

1.0 INTRODUCTION

1.1 Objective

The purpose of the 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 are also described in the OAD.

The science basis of an algorithm is described in a corresponding Algorithm Theoretical Basis Document (ATBD). 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. There is a general, but not strict, one-to-one correspondence between OAD and ATBD documents.

1.2 Scope

The scope of this document is limited to the description of the core operational algorithms required to create the VIIRS Ground Track Mercator (GTM) Imaging Band (I-Band) Imagery EDR and the VIIRS GTM Moderate Band (M-Band) Imagery EDR. The theoretical basis for these algorithms was developed by Raytheon with no ATBD as a reference document.

1.3 References

1.3.1 Document References

The science and system engineering documents relevant to the algorithms described in this OAD are listed in Table 1.

Table 1: Reference Documents

Document Title	Document Number/Revision	Revision Date
Joint Polar Satellite System (JPSS) Algorithm Specification Part 26	474-00448-01-24_JPSS-SRS-Vol-I-Part-26 474-00448-02-24_JPSS-DD-Vol-2-Part-26 474-00448-03-24_JPSS-OAD-Vol-III-Part-26 474-00448-04-24_JPSS-SRSPF-Vol-IV-Part-26	Latest
NPP Command and Telemetry (C&T) Handbook	D568423 Rev. C	30 Sep 2008

Document Title	Document Number/Revision	Revision Date
Operational Algorithm Description Document for Common Geolocation	474-00091	Latest
Operational Algorithm Description Document for VIIRS Near Constant Contrast (NCC) Imagery Environmental Data Records (EDR)	474-00060	Latest
JPSS Program Lexicon	470-00041	Latest
NGST/SE technical memo – NPP_VIIRS_GTM_Imagery_Handling_of_Bad_Detector_Data_Rev_A	NP-EMD.2006.510.0079 Rev A	16 Nov 2006

1.3.2 Source Code References

The science and operational code and associated documentation relevant to the algorithms described in this OAD are listed in Table 2. IDPS was not provided source code or test data for this algorithm. All source code and test data shown in Table 2 were developed by Raytheon.*

Table 2: Source Code References

Reference Title	Reference Tag/Revision	Revision Date
Initial OAD (Build 1.3)	OAD Rev ---	30 Jun 2004
VIIRS GTM Imagery EDR operational software	Build I1.4 (ECR 480)(OAD Rev A1)	04 Jan 2007
VIIRS GTM Imagery EDR operational software	Build I1.5 (OAD Rev A3)	27 Apr 2007
VIIRS GTM Imagery EDR operational software	Build I1.5.x.1 (OAD Rev A8)	21 Nov 2007
VIIRS GTM Imagery EDR operational software	Build I1.5.x.1 (OAD Rev A9)	15 Sep 2008
ACCB	OAD Rev A	13 Mar 2009
VIIRS GTM Imagery EDR operational software (PCR024523)	Build Sensor Characterization SC-13 (OAD Rev B4)	23 Aug 2010
PCR026647 (OAD update for ADL)	(OAD Rev B6)	26 Sep 2011

2.0 ALGORITHM OVERVIEW

2.0.1 GTM Background

The purpose of the VIIRS GTM Imagery algorithm is to map VIIRS Imaging (I) channel and Moderate (M) channel data onto a GTM layout. The GTM layout is a grid of pixels, where rows are at right angles to the ground track and columns are parallel to the ground track. This GTM layout does not have the “bow-tie” effect. The GTM Imagery EDR products are primarily used for visual snow/ice analysis and to display for human viewing.

Similar to Space Oblique Mercator (SOM), the GTM is not a map projection, i.e., it does not have an exact set of unchanging transformation equations. Rather, a numerical integration process allows for a latitude and longitude calculation of a row/column (X, Y) position on the map plane, or vice versa. With SOM, there are a finite number of map planes, based on numerically integrated orbit paths. The SOM map plane for an orbit path is based on the parameters of a model orbit, followed by a numerical integration. With GTM, the actual ground track of the spacecraft establishes the map plane; consequently, the map plane is different for every orbit. The GTM map has an advantage of always having the ground track in the center of the map plane. Furthermore, multiple granules of satellite data can be concatenated without having to switch from one orbit path map to another.

2.0.2 GTM Map Description

JPSS creates two kinds of GTM maps: Fine and Coarse. The Fine GTM map has a pixel-center spacing of 375 meters, which is close to the nadir sample distance of the VIIRS IMG resolution data. The Coarse GTM map has a pixel-center spacing of 750 meters, which is close to the nadir sample distance of the VIIRS MOD resolution data. The pixel spacing along the track direction is equal to the pixel spacing in the cross track direction. Because of these features, the GTM map is both conformal and equal area. The maximum variation, in both conformity and area per pixel, is about one percent (a variation which is a tiny fraction smaller than the SOM map). The X-coordinate on the GTM map increases in the direction of spacecraft motion along the ground track. The X axis is precisely on the ground track. The Y axis of the map is at a right angle to the X axis. That is, the rows of the GTM map are always at an exact right angle to the ground track. The time attached to each row of the map is the time the spacecraft passes over the nadir point of that row. See Figure 1: GTM Map Attributes

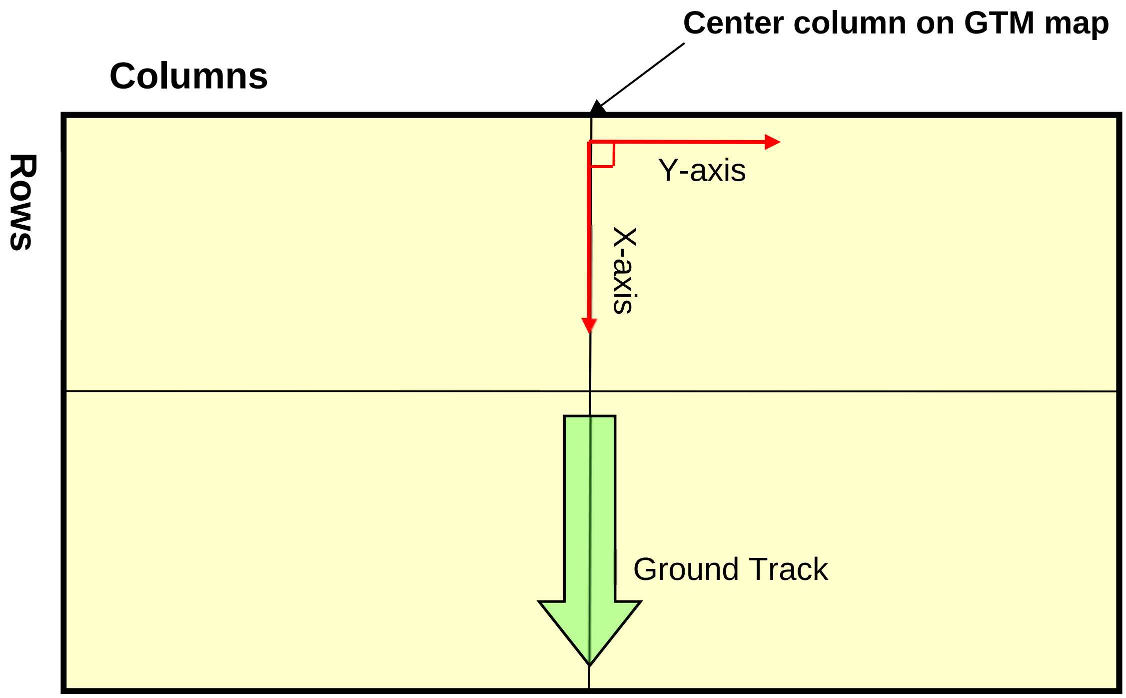


Figure 1: GTM Map Attributes

90-kilometer GTM swath was chosen to accommodate a maximum satellite altitude of 850 kilometers. There are 1541 rows and 8241 columns in the fine resolution GTM layout. The row size was chosen to accommodate the minimum altitude (maximum distance of a granule) of the satellite. There can be a variable number of empty columns on the edges of the swath, due to a larger area of the Earth's surface seen near the poles and less near the equator. Rows pull together slightly at the swath edges due to Earth curvature and horizontal size of the GTM swath. There are also a variable number of empty rows at the bottom of the GTM rectangle, due to the fixed horizontal sample distance and forward ground motion of the spacecraft. This layout allows concatenation of an unlimited number of EDR granules without any discontinuities, even at the poles.

The center column of the coarse map exactly follows the center column of the fine map. The pixel centers of the coarse map center column are the same as every other pixel of the fine map center column. Similarly, the pixel centers to the left and right of the center column on the coarse map have the same centers as every other pixel on the corresponding row of the fine map. In Figure 2: Fine Map Pixels with Emphasized Coarse Pixels each dot represents a pixel center on the GTM Fine Map. The emphasized dots represent coarse pixels on the Fine Map.

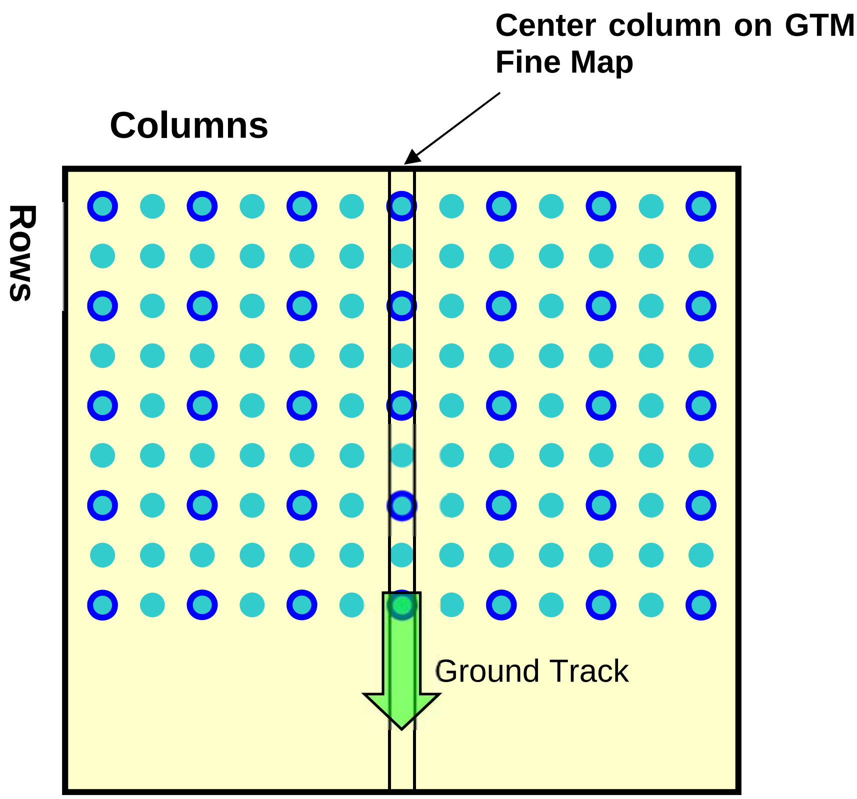


Figure 2: Fine Map Pixels with Emphasized Coarse Pixels

Even though the characteristics of the VIIRS sensor have been used to establish the parameters of the GTM maps, any kind of data can be remapped to the GTM maps. This makes it possible to form matching overlays from any number of data sources.

2.0.3 GTM Processing Overview

The first step in creating the GTM map data for an JPSS granule is to calculate the ground track by getting the ephemeris from the SDR GEO input. The basis of this ground track data is the ephemeris data reported by the GPS sensor on the spacecraft. This data comes down in the Ephemeris and Attitude data packets (these packets are also called "spacecraft diary" packets). Notice the direction of ground track motion is in the ECR system of the rotating Earth. This means the direction of ground track motion accounts for the rotation of the Earth as well as the inertial motion of the spacecraft. Sub-functions are used to calculate the ground track points for the start time and end time of the granule. The sub-functions are also used to space the rows of the coarse and fine GTM map as close to 375 meters as possible, and to put the center of the map precisely on the ground track. For the present granule size of about 85.752 seconds, there are about 1536 rows for Imaging resolution and 768 rows for Moderate resolution. The number of rows varies slightly because the granule size is a fixed number of seconds, and the ground track speed of the spacecraft varies slightly. The maximum variation in row spacing at nadir is 375 meters, +/- about 0.7 meters. Careful location of the first and last row in one granule means the GTM map of one granule always precisely concatenates with the GTM maps of the neighboring granules.

In summary, the centers of each row of the GTM map, the ground track pixels, are located by equal distance spacing of the pixels precisely on the ground track.

Once the locations of the ground track column pixels are established, it is possible to calculate the locations of the pixels along the rows by a simple application of spherical trigonometry. The geodetic latitude and longitude of the center pixel are used along with the radius of the Earth at that geodetic latitude. The direction from the center pixel to another pixel in the row is exactly 90 degrees to the left or right of the direction of ground track motion, which means the row is at an exact right angle to the ground track. All the pixels in one row of the map are theoretically at the same time as the center pixel, so there is no spacecraft motion or Earth rotation to account for along each row. The fact that the Earth is not an exact sphere is not a problem. The objective here is a reproducible map where there is a "one-to-one and onto" relationship between the surface of the Earth and the GTM map.

In summary, great circle distance and spherical trigonometry, location of the pixel in the center of each row, and direction of ground track motion are used to establish the location of each pixel along the row.

All of these calculations can be exactly reproduced because the entire process is based on a relatively small, single set of data: 1) the ephemeris data recorded by the GPS sensor on the spacecraft during the time span of the granule, 2) the deterministic granule boundaries (start and stop time of each granule) of the spacecraft, and 3) the 375 meter Earth surface distance between each pixel of the Fine GTM map.

2.0.4 Additional GTM Processing Details

The operational software only does full calculations for every 10th row and column, and then does quadratic interpolation of the pixels between. So, the calculation of a full set of latitudes and longitudes for a map is a relatively fast process.

The process of converting row and column to latitude and longitude, and vice versa, can be done by two methods. Method 1 is based on the fast search of a full set of geolocation data for the GTM map. Method 2 works from only the ground track data and works by an iterative search of the ground track, followed by a spherical trigonometry calculation along the row. Method 2 is slightly faster and the difference between the results is always less than one meter (the size of floating point round-off to 32 bits). The Nearest Neighbor method is used for filling pixels in order to preserve contrast and sharpness for human viewing. If full geolocation accuracy of the Sensor Data Record (SDR) is needed, the SDR should be used and not the GTM Imagery EDR.

Based on mode (day, mixed, or night) of the granule, data from either two or five imaging resolution channels are mapped onto the GTM map. In other words, two or five separate EDRs are created along with geolocation data for a given granule. Radiance and reflectance values for channels I1 through I3, along with radiance and brightness temperature values for channels I4 and I5, are processed for “day” and “mixed” mode granules. Radiance and brightness temperature values for channels I4 and I5 are processed for “night” mode granules. Currently, only six of the 16 moderate resolution channels are mapped onto the GTM map. The selection of which six channels to create is determined by a configuration file (see Section 2.3.7). It is the responsibility of the operator to select the appropriate bands for the mode of the granule day, mixed, or night) that is to be processed. Six EDRs are created using the short names VIIRS-M1ST-IMG-EDR through VIIRS-M6TH-IMG-EDR. The EDR metadata can be read to determine the Band ID used to generate the EDR. The Day Night Band (DNB) is processed by the Near Constant Contrast (NCC) Imagery algorithm to produce an EDR mapped to GTM. See the

Operational Algorithm Description Document for VIIRS Near Constant Contrast (NCC) Imagery Environmental Data Records (EDR), 474-00060, for more information. VIIRS GTM Imagery EDRs are not corrected for height.

2.1 GTM Imagery Base Algorithm Description

2.1.1 Interfaces

2.1.1.1 Inputs

Inputs are defined for the derived algorithms in their respective sections.

2.1.1.2 Outputs

Outputs are defined for the derived algorithms in their respective sections.

2.1.2 Algorithm Processing

2.1.2.1 Main Module - ProEdrViirsGtmImagery.h (template class)

This is the main GTM Imagery template class and implements the methods common to all of the GTM Imagery algorithms. When instantiated, it derives from one of the auto-generated GTM Imagery classes (IBand, MBand, or NCC) which in turn inherit from ProCmnAlgorithm. This class is used by the ProEdrViirsGtmIBandImagery, ProEdrViirsGtmMBandImagery, and ProEdrViirsGtmNccImagery derived classes. The inherited method applyAlgorithm() controls the flow of the GTM Imagery EDR code.

See the Operational Algorithm Description Document for VIIRS Near Constant Contrast (NCC) Imagery Environmental Data Records (EDR), 474-00060, for more information on the NCC algorithm.

2.1.2.1.1 doProcessing

All processing related routines are called from this method. This method is the main processing routine. It creates the I-Band, M-Band, or NCC Imagery products (EDRs and Geolocation dataset) for a given granule.

NOTE: The NCC Imagery part of the GTM Imagery algorithm does not actually produce the EDR. This part of the algorithm simply performs the mapping of the DNB SDR into a GTM moderate resolution sized temporary buffer. Then the NCC Imagery algorithm main routine is performed to convert the DNB radiances into the Near Constant Contrast Imagery EDR. See the NCC Imagery OAD, 474-00060, for more information.

2.1.2.1.2 fillOutputStructures

This pure virtual method fills the output geolocation and imagery EDR data structures (or the temporary SDR buffer for NCC Imagery). The derived algorithms must implement this function.

2.1.2.1.3 calculateSdrPixelLocations

This pure virtual method is used to calculate the corresponding SDR pixel location for each pixel in the Imaging resolution GTM granule. The derived algorithms must override this function as described below.

2.1.2.2 Derived Algorithms

2.1.2.2.1 doProcessing in the derived algorithms

Invoke the routines for creating the I-Band, M-Band, or NCC Imagery EDRs and the Geolocation EDR for a given granule.

- I-Band, M-Band, NCC Imagery invokes bld_gtm_grndtrk_data() to build the ground track
- I-Band, M-Band, NCC Imagery then invokes bld_full_mod_gtm() to build the moderate geolocation data
- Only I-Band then invokes bld_full_img_gtm() to convert the moderate resolution data into the imagery resolution data
- I-Band, M-Band, NCC Imagery then invokes calculateSdrPixelLocations() and fillOutputStructures() to map the SDR pixels to the EDR and create the output items

2.1.2.2.2 fillOutputStructures in the derived algorithms

Each of the derived algorithms must implement this function to copy the EDR array data from the appropriate structures. In general each algorithm performs the following steps:

For each output EDR perform the following steps:

Set up pointers to the Next and Previous granule data, or NULL if the Next or Previous data does not exist.
CALL the appropriate fillArray() method.

The fillArray() method is a template for the array type being copied in the EDR data (or SDR temporary buffer for NCC Imagery) and takes in as parameters the size of the data buffer being copied. The I-Band and M-Band classes use Float32 arrays and the NCC Imagery uses a Float32 array for radiance and an unsigned character array for the quality bits.

Copy the temporary pixel locations structure to the GEO output item.

This step is necessary to convert the internal array of structures into a DMS compatible structure of arrays.

2.1.2.2.3 calculateSdrPixelLocations in the derived algorithms

This method calculates the corresponding SDR pixel location for each pixel in the GTM granule. These values are written to a temporary array and are later written to the output geolocation structure.

This method is a pure virtual method that must be instantiated by each of the derived algorithms to operate on the appropriately sized data structures.

1. Create two grid point conversion objects for converting grid points between the primary granule's grid and a neighboring granule's grid. Initialize for a granule grid type (polar

stereographic). The four input grid points are arbitrarily chosen to form a square inside the equator on the polar stereographic grid calculated by the SDR process.

2. Loop over GTM rectangles(20x20 pixel rectangles) of the Fine resolution geolocation data and fill a temporary array with SDR pixel locations
3. Convert a grid position to an SDR pixel location by calling the grid_to_(IMG/MOD/DNB)sdrPixel() C Function

    CALL the appropriate grid to SDR pixel method:grid_to_modSDRpixel(grid row, grid column, &pixelRow, &pixelCol)
    grid_to_imgSDRpixel(grid row, grid column, &pixelRow, &pixelCol)
    grid_to_dnbSDRpixel(grid row, grid column, &pixelRow, &pixelCol)

    If the method returns an error (pixel search failed or not a valid warning code), then send a debug message and return PRO_FAIL.

    If the method returns a warning (pixel not in the SDR), then

    1. Map the pixel based on the errorCode using the appropriate grid point conversion object created above.
    2. CALL the appropriate grid to SDR pixel method again.
    3. Fill that pixel with a flag value based on the second errorCode returned by the grid to SDR pixel method.

    If the pixel is in the primary SDR granule, then

    1. Check for boundary row pixel trimmed values and convert them to the previous or next granule. Otherwise, the pixel is in the primary granule and the row and column have been properly calculated.
    2. Store the pixel location.

2.1.2.3 Reuse C Functions

2.1.2.3.1 bld_gtm_grndtrk_data

This method builds the structure that contains the GTM ground track data for every 10th row.

1. Calculate number of actual rows in the GTM grid, number of rows of geolocation data, and number of rows for the temporary grid data.
    a. Invoke gtm_grndtrk_ptr() to find the nadir point data for the end of the granule.
    b. Invoke gtm_grndtrk_ptr() to find the begin track point from begin time and gtm_ephem.
    c. Calculate the distance from granule begin nadir point to granule end nadir point. The earth radius is derived for the average latitude.
    d. Adjust the distance between rows so that a GTM row nadir point will be exactly on the granule end nadir point.
    e. Then determine the ground track points until you reach one past the granule end point, there should be an odd number of ground track points. We enter this loop as 1. So, the first pass value of idx_trk_prv is 0.
    f. target_dist in this loop is the target distance from the point being generated to the beginning of the granule. The point being generated is moved back and forth until the distance is correct. target_dist is the 10 row distance times the ground track index of the point being generated. The objective of this loop is generate points that are exactly each target distance from the beginning of the granule.

Table 3: bld_gtm_grndtrk_data Parameter Definitions

Parameter	Type	I/O	Description
gran_bgnTAI	Double	I	granule begin boundary time
gran_endTAI	Double	I	granule end boundary time
dist10rows	Double	I	distance in meters for 10 GTM rows
gtm_ephem	GTM_ephemeris_data_type*	I	GTM SDR ephemeris data.
gtmtrk_data	gtm_ground_track_data_type*	O	GTM ground track data for use in generating* GTM geolocation data by 20 row interpolation.

2.1.2.3.2 bld_full_mod_gtm

This method builds the full set of geolocation data for an MOD GTM granule from GTM ground track data.

1. Initialize the MOD GTM geolocation data to arbitrary value.
2. Check to make sure the ground track data has a distance of ~7500 meters.
3. Make sure the input data has compatible number of rows.
4. Put the ground track data into every 10th row of the MOD GTM data.
5. The full number of GTM rows along the ground is larger than the actual number of GTM rows. Because of the way that idx_frow was incremented, it is now too large. We want a number which is 1 more than the last index value used, and that is 9 less than its present value.
6. Generate lat-lon data, and map coordinate data, for every 10 rows and 10 columns. This data will be used to interpolate map coordinates for all points in the full GTM.
7. For all the column locations in this row, we will use the Earth Radius at the latitude of the low nadir point. By doing this we will absolutely ensure granule to granule continuity.

Table 4: bld_full_mod_gtm Parameter Definitions

Parameter	Type	I/O	Description
gtmtrk_data	gtm_ground_track_data_type*	I	Ground track data for ~7500 meter spacing
hem_mds	mds_type*	I	Map Data Set for a hemisphere
fm_gtm	full_mod_gtm_type*	O	filled out MOD GTM geolocation data

2.1.2.3.3 bld_full_img_gtm

This method uses the MOD GTM geolocation data to create the IMG GTM geolocation data.

1. Initialize the IMG GTM structure with fill
2. Copy the Map Data Set.
3. Copy data from MOD to IMG. The time and map coordinate data will be copied from the MOD data to the IMG data. The MOD data is copied to every other column on every other row of the IMG data. The lat-lon data is not copied from the MOD data structure, only the map coordinates are copied. Wherever needed, the IMG map coordinates will be interpolated. After all the map coordinates are interpolated the lat-lon data will be calculated from the map coordinates.

Table 5: bld_full_img_gtm Parameter Definitions

Parameter	Type	I/O	Description
fm_gtm	full_mod_gtm_type*	I	MOD GTM geolocation data structure
fi_gtm	full_img_gtm_type*	O	IMG GTM geolocation data structure

2.1.2.3.4 gtm_grndtrk_pt

This method calculates the geoditic latitude, longitude, and azimuth of ground track motion, for the satellite nadir point for each input item.

1. Find the two ephemeris reports that the input time is between
2. Do a linear interpolation of ECR position and velocity
3. Obtain the information that is needed for output.

Table 6: gtm_grndtrk_pt Parameter Definitions

Parameter	Type	I/O	Description
trkTAI	Double	I	desired time of the output data
Gtm_ephem	Gtm_ephemeris_data_type*	I	GTM ephemeris data structure
dlat	Double*	O	ground track point geodetic latitude
lon	Double*	O	ground track point longitude.
trkazm	double*	O	ground track azimuth of motion.

2.1.2.3.5 short_dist

This function is used to calculate the distance between two points that are less than 10 kilometers apart.

Table 7: short dist Parameter Definitions

Parameter	Type	I/O	Description
c_lat	Double	I	first point latitude
c_lon	Double	I	first point longitude
a_lat	Double	I	second point latitude
a_lon	Double	I	second point longitude
arclen	double*	O	great circle arc length between points

2.1.2.3.6 azm_sidb

This function calculates bearing (in radians) and distance (earth central angle in radians) from point C to point A, based on C and A latitude/longitude of points. Table 8 shows the azm_sidb parameter definitions.

Table 8: azm_sidb Parameter Definitions

	Type	I/O	Description
Clat	Double	I	Latitude of the first point, in radians positive north.
Clon	Double	I	Longitude of the first point, in radians positive east.
Alat	Double	I	Latitude of the second point, in radians positive north.
Alon	Double	I	Longitude of the second point, in radians positive east.
Azimuth	double*	O	Bearing from C to A, in radians.
side_b	double*	O	Distance from C to A, in radians.

NOTE: "Side_b" is a distance measured as an angle at the center of the earth. The angle goes from point C to point A, along the great circle between, and the vertex is the center of the earth. The maximum value of "side_b" is pi.

NOTE: "Bearing" is the standard definition. "Bearing" in radians goes from zero to 2pi, with North=0, East=pio2, South=pi, West=trepio2. "Bearing" and "azimuth" are the same thing.

CONSTRAINTS: All inputs for latitude and longitude are positive north positive east.

-pio2 < latitude < pio2, -pi < longitude < pi

Making sure the inputs are in a legal range is the responsibility of the calling program.

2.1.2.3.7 target_pt

Below is a discussion of function target_pt, which does all of the spherical trig calculations described in Section 2.0, Algorithm Overview. Figure 3: Target_pt Function Calculations Diagram shows a diagram of how these calculations are made.

The spherical trigonometry used to calculate a second point from a first point is based on latitude and longitude of the start point, radius of the sphere at the start point, direction from start point to target point, and the laws of spherical trigonometry. The oblique spherical triangle used has the vertices: vertex A is the target point (unknown lat/lon), vertex B is the North Pole, and vertex C is the start point.

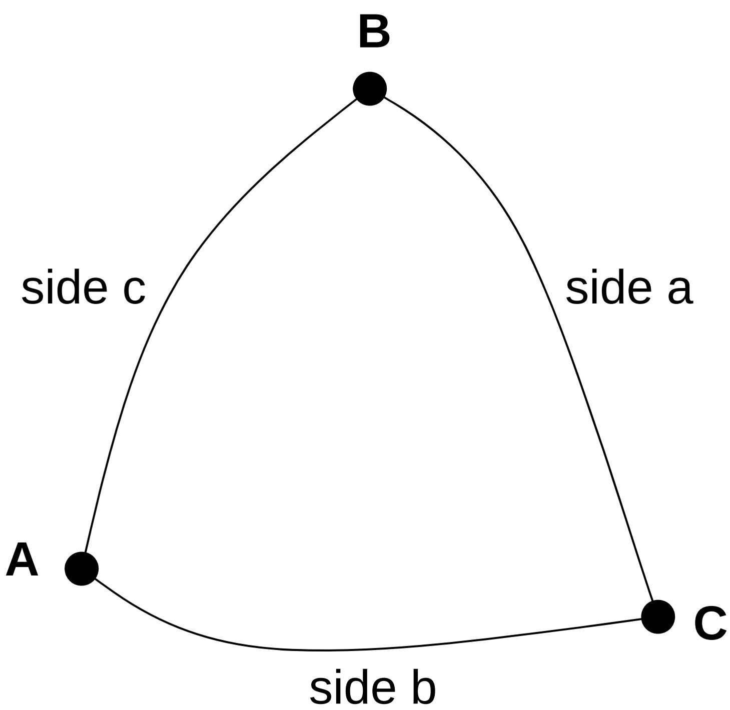


Figure 3: Target_pt Function Calculations Diagram

In spherical trigonometry, each side of the triangle is a great circle arc between the two vertices. The length of the side of a triangle is the angle measured at the center of the sphere, measured along the great circle arc from one vertex to the other.

From the input azimuth, it is determined that the target point is East or West of the start point. It is also known that side b is the distance from start point to target point, divided by the radius of the Earth at the start point.

Side a of this triangle is the longitude line from the North Pole to the start point. So, side a is equal to 90 degrees minus the latitude of the start point.

Angle C is directly determined by the azimuth from the start point to the target point. Angle C has to be adjusted depending on whether the target point is East or West of the start point.

Then the cosine of side c is determined from the Law of Cosines for Oblique Spherical triangles:

$$\cos(c) = \cos(a) \bullet \cos(b) + \sin(a) \bullet \sin(b) \bullet \cos(C)$$

Then side c is determined from the arc cosine function, and the latitude of the target point (one output of the function) is just (90 - side c).

Then the cosine of angle B is determined by again applying and rearranging the Law of Cosines for Oblique Spherical Triangles:

$$\cos(B) = \frac{\cos(b) - \cos(c) \bullet \cos(a)}{\sin(c) \bullet \sin(a)}$$

Then angle B is determined from the arc cosine function and the longitude of the target point (the second output of the function) is the start point longitude plus or minus angle B, depending whether the target point is East or West of the start point.

This function has the inputs of a latitude and longitude start point, plus distance and direction to a second point. Outputs are latitude and longitude of the second point. Table 9 shows the target_pt parameter definitions.

Table 9: target_pt Parameter Definitions

Parameter	Type	I/O	Description
Clat	Double	I	Latitude of the start point.
Clon	Double	I	Longitude of the start point.
side_b	Double	I	Distance to the second point.
Azimuth	Double	I	Azimuth to the second point.
Alat	Double*	O	Latitude of the second point.
Alon	Double*	O	Longitude of the second point.

NOTE: All latitudes and longitudes are in radians, positive North and positive East.

-pio2 < Latitude < pio2, -pi < Longitude < pi

NOTE: side_b is an angle measured at the center of the earth.

CONSTRAINTS: It is the responsibility of the calling program to keep inputs inside legal ranges.

NOTE: This function resolves a triangle on the surface of a sphere using laws from spherical trigonometry. Vertices of the triangle are: Point A (target point), Point B (North Pole), and Point C (start point).

2.1.2.3.8 grid_to_imgSDRpixel

This function takes in an interpolation grid position and the geolocation data for an Imaging Resolution SDR granule, and then converts location to an SDR pixel location (row and column). Table 10 shows the grid_to_imgSDRpixel parameter definitions.

Table 10: grid_to_imgSDRpixel Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	Input row on granule interpolation grid.
gcol	Double	I	Input column on granule interpolation grid.
iRect	ViirsGeoRctnglType&	I	Rectangle parameters.
vI_growcol	viirs_SDR_IMG_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.3.9 grid_to_modSDRpixel

This function takes in an interpolation grid position and the geolocation data for a Moderate Resolution SDR granule, and then converts location to an SDR pixel location (row and column). Table 11 shows the grid_to_modSDRpixel parameter definitions.

Table 11: grid_to_modSDRpixel Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	Input row on granule interpolation grid.
gcol	Double	I	Input column on granule interpolation grid.
iRect	ViirsGeoRctnglType&	I	Rectangle parameters.
vM_growcol	viirs_SDR_MOD_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.3.10 grid_to_dnbSDRpixel

This function takes in an interpolation grid position and the geolocation data for a Day Night Band SDR granule, and then converts location to an SDR pixel location (row and column). Table 12 shows the grid_to_imgSDRpixel parameter definitions.

Table 12: grid_to_dnbSDRpixel Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	Input row on granule interpolation grid.
gcol	Double	I	Input column on granule interpolation grid.
iRect	ViirsGeoRctnglType&	I	Rectangle parameters.
vD_growcol	viirs_SDR_DNB_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.3.11 rp_g2imgpix

This function refines pixel location of the input grid position in the VIIRS IMG granule. Function grid_to_imgSDRpixel takes in grow, gcol and determines the pixel (iprow, ipcol) closest to that input. This function refines that location to a fraction of a pixel. Table 13 shows the rp_g2imgpix parameter definitions.

Table 13: rp_g2imgpix Parameter Definitions

Parameter	Type	I/O	Description
Grow	Double	I	Input row on granule interpolation grid.
Gcol	Double	I	Input column on granule interpolation grid.
Iprow	Int	I	Row of pixel closest to input point.
Ipcol	Int	I	Column of pixel closest to input point.

Parameter	Type	I/O	Description
Iprow_bgn	Int	I	First row of scan containing iprow.
Iprow_end	Int	I	Last row of scan containing iprow.
vI_growcol	viirs_SDR_IMG_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.3.12 rp_g2modpix

This function refines pixel location of the input grid position in the VIIRS MOD granule. Function grid_to_modSDRpixel takes in grow, gcol and determines the pixel (iprow, ipcol) closest to that input. This function refines that location to a fraction of a pixel. Table 14 shows the rp_g2modpix parameter definitions.

Table 14: rp_g2modpix Parameter Definitions

Parameter	Type	I/O	Description
Grow	Double	I	Input row on granule interpolation grid.
Gcol	Double	I	Input column on granule interpolation grid.
Iprow	Int	I	Row of pixel closest to input point.
Ipcol	Int	I	Column of pixel closest to input point.
Iprow_bgn	Int	I	First row of scan containing iprow.
Iprow_end	Int	I	Last row of scan containing iprow.
vM_growcol	viirs_SDR_MOD_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.3.13 rp_g2dnbdpix

This function refines pixel location of the input grid position in the VIIRS DNB granule. Function grid_to_dnbSDRpixel takes in grow, gcol and determines the pixel (iprow, ipcol) closest to that input. This function refines that location to a fraction of a pixel. Table 15 shows the rp_g2dnbdpix parameter definitions.

Table 15: rp_g2dnbdpix Parameter Definitions

Parameter	Type	I/O	Description
Grow	Double	I	Input row on granule interpolation grid.
Gcol	Double	I	Input column on granule interpolation grid.
Iprow	Int	I	Row of pixel closest to input point.
Ipcol	Int	I	Column of pixel closest to input point.
Iprow_bgn	Int	I	First row of scan containing iprow.
Iprow_end	Int	I	Last row of scan containing iprow.
vD_growcol	viirs_SDR_DNB_growcol_type*	I	Grid locations of all granule pixels.
Pixrow	Double	O	Floating point pixel row.
Pixcol	Double	O	Floating point pixel column.
Lerr	Int	O	Returned error code.

2.1.2.3.14 grid_to_latlon

This function converts a row column position from an MDS to a latitude and longitude. Table 16 shows the grid_to_latlon parameter definitions.

Table 16: grid_to_latlon Parameter Definitions

Parameter	Type	I/O	Description
Row	Double	I	Row coordinate on the grid.
Col	Double	I	Column coordinate on the grid.
Imds	mds_type*	I	Pointer to the input map data set structure.
Rlat	double*	O	Pointer to the output latitude.
Rlon	double*	O	Pointer to the output longitude.
err_string	char*	O	Pointer to a 256 byte string.
Err	Int	O	Returned error code.

2.1.2.3.15 latlon_to_grid

This function converts a latitude and longitude to a position on a Map Data Set grid. Table 17 shows the latlon_to_grid parameter definitions.

Table 17: latlon_to_grid Parameter Definitions

Parameter	Type	I/O	Description
Rlat	Double	I	Input latitude.
Rlon	Double	I	Input longitude.
Imds	mds_type*	I	Pointer to the input map data set structure.
Row	double*	O	Pointer to the row coordinate on the grid.
Col	double*	O	Pointer to the column coordinate on the grid.
Err_string	char*	O	Pointer to a 256 byte string.
Err	Int	O	Returned error code.

2.1.2.3.16 earth_radius_D

This function computes radius of the Earth, in kilometers, from the geodetic latitude. Table 18 shows the earth_radius_D parameter definitions.

Table 18: earth_radius_D Parameter Definitions

Parameter	Type	I/O	Description
Rlat	Double	I	Geodetic latitude, radians, positive north.
Radius	Double	O	Returned radius of the earth in kilometers.

2.1.2.3.17 grid2img_pix_wiscan

This function takes in the map grid position being searched for, data about the SDR, and data about the place to start the search. It returns the closest pixel within the scan where the search starts. "wiscan" means "within scan**".**

Table 19: grid2img_pix_wiscan Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	interpolation grid row
gcol	Double	I	interpolation grid column
vl_growcol	viirs_SDR_IMG_growcol_type*	I	granule grid locations of all pixels
kcan	Int	I	scan where search starts
in_prowl	Int	I	pixel row where search starts
in_pcol	Int	I	pixel col where search starts
pixrow	Double *	O	closest SDR pixel row number
pixcol	Double *	O	closest SDR pixel column number

2.1.2.3.18 grid2mod_pix_wiscan

This function takes in the map grid position being searched for, data about the SDR, and data about the place to start the search. It returns the closest pixel within the scan where the search starts. "wiscan" means "within scan".

Table 20: grid2mod_pix_wiscan Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	interpolation grid row
gcol	Double	I	interpolation grid column
vM_growcol	viirs_SDR_MOD_growcol_type*	I	granule grid locations of all pixels
kcan	Int	I	scan where search starts
in_prowl	Int	I	pixel row where search starts
in_pcol	Int	I	pixel col where search starts
pixrow	Double *	O	closest SDR pixel row number
pixcol	Double *	O	closest SDR pixel column number

2.1.2.3.19 grid2dnb_pix_wiscan

This function takes in the map grid position being searched for, data about the SDR, and data about the place to start the search. It returns the closest pixel within the scan where the search starts. "wiscan" means "within scan".

Table 21: grid2dnb_pix_wiscan Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	interpolation grid row
gcol	Double	I	interpolation grid column
vD_growcol	viirs_SDR_DNB_growcol_type*	I	granule grid locations of all pixels
kcan	Int	I	scan where search starts
in_prowl	Int	I	pixel row where search starts
in_pcol	Int	I	pixel col where search starts
pixrow	Double *	O	closest SDR pixel row number
pixcol	Double *	O	closest SDR pixel column number

2.1.3 Graceful Degradation

2.1.3.1 Graceful Degradation Inputs

None.

2.1.3.2 Graceful Degradation Processing

None.

2.1.3.3 Graceful Degradation Outputs

None.

2.1.4 Exception Handling

Missing sensor data caused by bad detectors are replaced during pre-processing of the GTM algorithm as follows:

For edge of scan, the radiance value of the adjacent pixel is copied into the missing pixel. For non-edge of scan, the radiances are averaged using the two adjacent pixels and copied into the missing pixel. Missing data points are left as the initialization fill value in the GTM Imagery EDRs.

Additionally, the VIIRS GTM Imagery software is designed to handle a wide variety of processing problems. Any exceptions or errors are reported to IDPS using the appropriate INF API.

2.1.5 Data Quality Monitoring

None.

2.1.6 Computational Precision Requirements

The GTM Imagery algorithm is a pass-through of the SDR radiance, reflectance, and brightness temperature data, and no change of precision occurs in the code. Geolocation (latitude and longitude) computations are done in 32-bit and 64-bit floating point precision. The final latitude and longitude values are accurate to one meter. All time computations are done in 64-bit floating point and 64-bit integer precision.

In order to speed GTM Imagery processing, an interpolation scheme is used to calculate grid data for the granule. The latitude and longitude and associated grid row and column values are calculated for every 10^{th} point in the GTM grid. Then quadratic interpolation is performed over 20x20 pixel rectangles within the GTM grid to obtain the full geolocation for the granule. The maximum error induced by the interpolation is one meter. Row times are calculated by performing linear interpolation between the granule boundary times.

2.1.7 Algorithm Support Considerations

DMS and INF must be running before execution of the GTM Imagery algorithm.

2.1.8 Assumptions and Limitations

No assumptions or limitations have been identified.

2.2 GTM Imagery I-Band Class Description

ProEdrViirsGtmIBandImagery is the main GTM I-Band Imagery process. It instantiates an I_Band instance of the ProEdrViirsGtmImagery template class, by deriving from the auto-generated IBand algorithm class, “AutoGeneratedProEdrViirsGtmIChannelImagery”, and calls the inherited applyAlgorithm() method.

The basic flow of the I-Band Imagery algorithm is depicted in Figure 4: Basic Processing Flow for the VIIRS I-Band Imagery EDR Inputs are the VIIRS SDRs (channels I1 through I5), VIIRS I-Band SDR grid data, and VIIRS I-Band sensor look angles. Outputs are the I-Band Imagery EDRs and geolocation data.

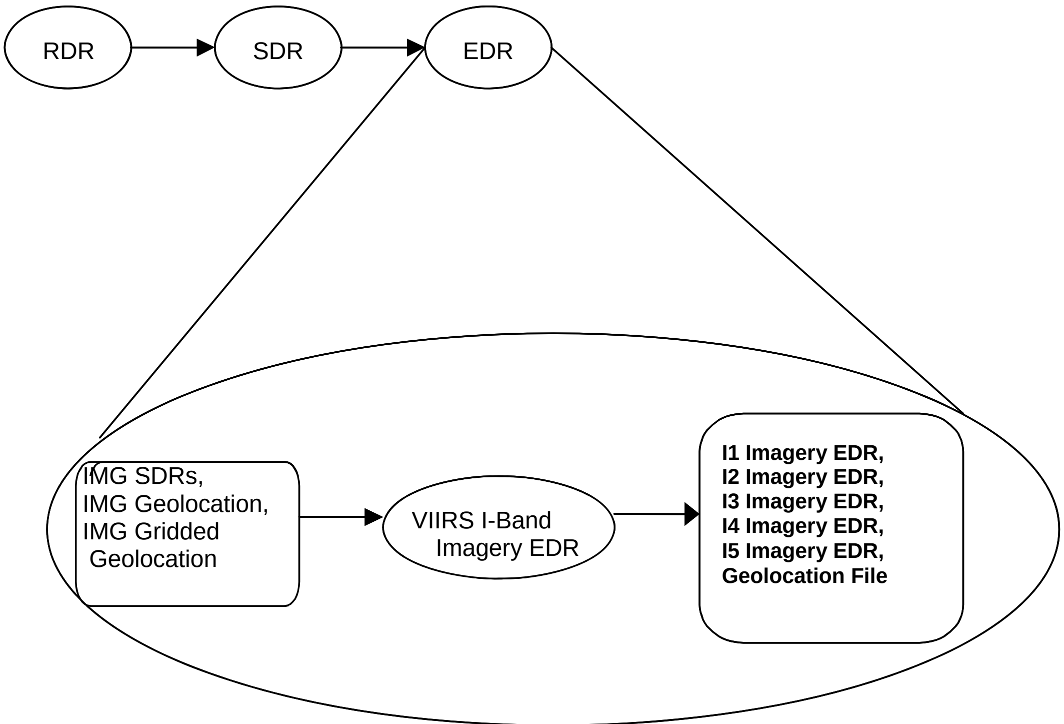


Figure 4: Basic Processing Flow for the VIIRS I-Band Imagery EDR

2.2.1 Interfaces

2.2.1.1 Inputs

VIIRS I-Band Imagery algorithm requires several types of data to perform mapping to the GTM layout, summarized in 474-00448-01-26_JPSS-SRS-Vol-I-Part-26, Table 3-1.

Table 22: VIIRS I-Band Imagery EDR Inputs

Inputs	Description	Reference Documents
VIIRS Band I1, I2, and I3 SDRs	Radiances and reflectance, granule boundary times and granule mode. Used for day and mixed mode granules.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS Band I4 and I5 SDRs	Brightness temperatures and radiances, granule boundary times and granule mode. Used for day, mixed and night mode granules.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS IMG Geolocation	Ellipsoid Geolocation data for every pixel in the granule.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS IMG Gridded Geolocation	Map grid row and column values for every pixel in the granule and the granule MDS.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
Band I1–I5 EDR DQTTs	Data Quality Threshold Tables used for performing data quality checks on the EDR outputs. These are optional inputs.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26

2.2.1.2 Outputs

VIIRS I-Band Imagery EDR outputs are summarized in 474-00448-01-26_JPSS-DD-Vol-I-Part-26, Table 3-1. The outputs are further defined in 474-00448-02-26_JPSS-DD-Vol-02-Part-26, Tables 5.1.1.1-1 through 5.1.1.18-1 [Imagery Output and Imagery QFs] and 5.1.1.22-1 and 5.1.1.23-1 [GEO Output and GEO QFs].. Note that the I1 – I3 band EDRs have a reflectance field where as the I4 and I5 band EDRs have a brightness temperature field. Latitude and longitude are calculated as the center of the GTM pixel. Solar angle, satellite angle, terrain height and satellite range are copied from the source SDR pixel.

Table 23: VIIRS I-Band Imagery EDR Outputs

Name	Description	Reference Documents
VIIRS Band I1, I2, I3, I4, I5 Imagery EDRs	VIIRS-I*-EDR contains [scaled] data fields, unscaled EDR products and Quality flags	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS IMG EDR Geolocation	VIIRS IMG EDR Geolocation Data	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26

2.2.2 Algorithm Processing

2.2.2.1 Main Module - ProEdrViirsGtmIBandImagery.cpp

This class is the implementation of the VIIRS Imaging Band Imagery algorithm that computes the I-Band Imagery EDRs mapped to the Ground Track Mercator (GTM) map. This class instantiates the ProEdrViirsGtmImagery template class deriving from the auto-generated IBand Imagery class, “AutoGeneratedProEdrViirsIChannelImagery”, which in turn derives from ProCmnAlgorithm.

The I-Band Imagery EDRs contain VIIRS imaging band data mapped onto a Ground Track Mercator (GTM) layout. Depending on the granule mode (DAY, MIXED, or NIGHT), two or five I-Band Imagery EDRs are created per granule. One geolocation dataset is also created per granule.

2.2.2.1.1 setupDataItems

This method implements the pure virtual base class method. It works in conjunction with auto-generated source code (AutoGeneratedProEdrViirsIChannelImagery) to perform processing related to the setup of input and output data items needed for VIIRS I-Band Imagery processing.

2.2.2.1.2 doProcessing

The algorithm does a pre-processing step using the 32-bit bad detector quality flags located in the SDR input. If the first detector is bad, the radiance value in [row+1] is copied into the current row. If the last detector is bad, the radiance value in [row-1] is copied into the current row. For other bad detectors, the before and after rows are averaged and placed into the current row.

2.2.2.1.3 initOutputDataItems

This method works with auto-generated source code (AutoGeneratedProEdrViirsIChannel Imagery) to initialize each output data item's DMS data buffer.

2.2.3 Graceful Degradation

2.2.3.1 Graceful Degradation Inputs

None.

2.2.3.2 Graceful Degradation Processing

None.

2.2.3.3 Graceful Degradation Outputs

None.

2.2.4 Exception Handling

Missing sensor data caused by bad detectors are replaced during pre-processing of the GTM algorithm as follows:

For edge of scan, the radiance value of the adjacent pixel is copied into the missing pixel. For non-edge of scan, the radiances are averaged using the two adjacent pixels and copied into the missing pixel. Missing data points are left as the initialization fill value in the GTM Imagery EDRs.

Additionally, the VIIRS GTM Imagery software is designed to handle a wide variety of processing problems. Any exceptions or errors are reported to IDPS using the appropriate INF API.

2.2.5 Data Quality Monitoring

None.

2.2.6 Computational Precision Requirements

The GTM Imagery algorithm is a pass-through of the SDR radiance, reflectance, and brightness temperature data, and no change of precision occurs in the code. Geolocation (latitude and longitude) computations are done in 32-bit and 64-bit floating point precision. The final latitude and longitude values are accurate to one meter. All time computations are done in 64-bit floating point and 64-bit integer precision.

In order to speed GTM Imagery processing, an interpolation scheme is used to calculate grid data for the granule. The latitude and longitude and associated grid row and column values are calculated for every 10^{th} point in the GTM grid. Then quadratic interpolation is performed over 20x20 pixel rectangles within the GTM grid to obtain the full geolocation for the granule. The maximum error induced by the interpolation is one meter. Row times are calculated by performing linear interpolation between the granule boundary times.

2.2.7 Algorithm Support Considerations

DMS and INF must be running before execution of the GTM Imagery algorithm.

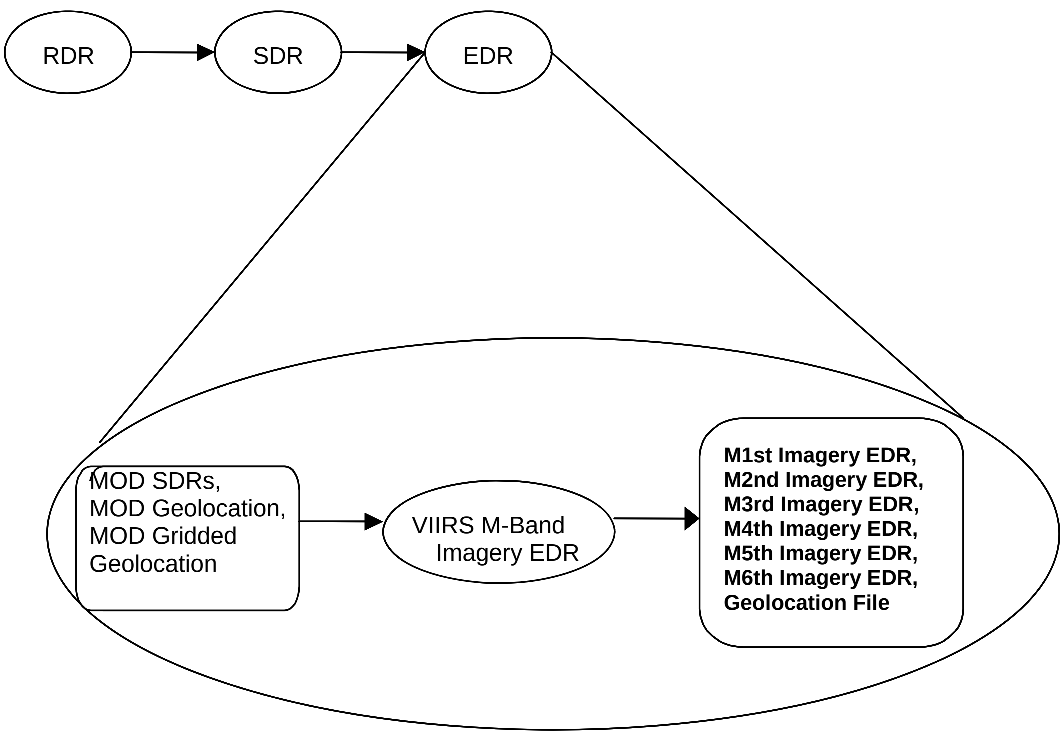
2.2.8 Assumptions and Limitations

No assumptions or limitations have been identified.

2.3 GTM Imagery M-Band Class Description

ProEdrViirsGtmMBandImagery is the main GTM M-Band Imagery process. It instantiates an M_Band instance of the ProEdrViirsGtmImagery template class, by deriving from the auto-generated MBand algorithm class, “AutoGeneratedProEdrViirsGtmMChannelImagery”, and calls the inherited applyAlgorithm() method.

The basic flow of the M-Band Imagery algorithm is depicted in Figure 5. Inputs are the VIIRS Mod SDRs, VIIRS Mod SDR grid data, and Mod VIIRS sensor look angles. Outputs are the M-Band Imagery EDRs and geolocation data.



2.3.1 Figure 5: Basic Processing Flow for the VIIRS M-Band Imagery EDR

2.3.1.1 Inputs

VIIRS M-Band Imagery algorithm requires several types of data to perform mapping to the GTM layout, summarized in 474-00448-01-26_JPSS-SRS-Vol-I-Part-26, Table 3-1.

Table 24: VIIRS M-Band Imagery EDR Inputs

Name	Description	Reference Documents
VIIRS Band M1 thru M6, M9, and M11 SDRs	Radiances and reflectance, granule boundary times and granule mode. Used for day and mixed mode granules.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS Band M7, M8, M10, and M12 thru M16 SDRs	Radiances and brightness temperatures, granule boundary times and granule mode. Used for day, mixed and night mode granules.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS MOD Geolocation	Ellipsoid Geolocation data for every pixel in the granule.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS MOD Gridded Geolocation	Map grid row and column values for every pixel in the granule and the granule MDS.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26

2.3.1.2 Outputs

VIIRS M-Band Imagery EDR outputs are summarized in 474-00448-01-26_JPSS-DD-Vol-I-Part-26, Table 3-1. The outputs are further defined in 474-00448-02-26_JPSS-DD-Vol-02-Part-26, Tables 5.1.2.1-1 through 5.1.1.17-1 [Imagery Output and Imagery QFs] and 5.1.2.21-1 and 5.1.2.22-1 [GEO Output and GEO QFs]. Note that the M1 – M6, M9, and M11 band EDRs have a reflectance field where as the M7, M8, M10, and M12 – M16 band EDRs have a brightness temperature field. Latitude and longitude are calculated as the center of the GTM pixel. Solar angle, satellite angle, terrain height and satellite range are copied from the source SDR pixel

Table 25: VIIRS M-Band Imagery EDR Outputs

Name	Description	Reference Documents
VIIRS Band M1 – M16 Moderate EDRs	VIIRS-M*-EDR contains [scaled] data fields, unscaled EDR products and Quality flags	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS Moderate EDR Geolocation	**VIIRS MOD EDR Geolocation Data**	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26

2.3.2 Algorithm Processing

2.3.2.1 Main Module - ProEdrViirsGtmMBandImagery.cpp

This class is the implementation of the VIIRS Moderate Band Imagery algorithm that computes the M-Band Imagery EDRs mapped to the Ground Track Mercator (GTM) map. This class instantiates the ProEdrViirsGtmImagery template class deriving from the auto-generated MBand Imagery class, "AutoGeneratedProEdrViirsMChannelImagery", which in turn derives from ProCmnAlgorithm.

The M-Band Imagery EDRs contain VIIRS moderate band data mapped onto a Ground Track Mercator (GTM) layout. Depending on the setting of the Mode keyword for the M-Band inputs in the configuration guide, six out of the 16 M-Band Imagery EDRs are created per granule. One geolocation dataset is also created per granule.

2.3.2.1.1 setupDataItems

This method implements the pure virtual base class method. It works in conjunction with auto-generated source code (AutoGeneratedProEdrViirsMChannelImagery) to perform processing related to the setup of input and output data items needed for VIIRS M-Band Imagery processing.

2.3.2.1.2 doProcessing

The algorithm does a pre-processing step using the 16-bit bad detector quality flags located in the SDR input. If the first detector is bad, the radiance value in [row+1] is copied into the current row. If the last detector is bad, the radiance value in [row-1] is copied into the current row. For other bad detectors, the before and after rows are averaged and placed into the current row.

2.3.2.1.3 initOutputDataItems

This method works with auto-generated source code (AutoGeneratedProEdrViirsMChannel Imagery) to initialize each output data item's DMS data buffer.

2.3.3 Graceful Degradation

2.3.3.1 Graceful Degradation Inputs

None.

2.3.3.2 Graceful Degradation Processing

None.

2.3.3.3 Graceful Degradation Outputs

None.

2.3.4 Exception Handling

Missing sensor data caused by bad detectors are replaced during pre-processing of the GTM algorithm as follows:

For edge of scan, the radiance value of the adjacent pixel is copied into the missing pixel. For non-edge of scan, the radiances are averaged using the two adjacent pixels and copied into the missing pixel. Missing data points are left as the initialization fill value in the GTM Imagery EDRs.

Additionally, the VIIRS GTM Imagery software is designed to handle a wide variety of processing problems. Any exceptions or errors are reported to IDPS using the appropriate INF API.

2.3.5 Data Quality Monitoring

None.

2.3.6 Computational Precision Requirements

The GTM Imagery algorithm is a pass-through of the SDR radiance, reflectance, and brightness temperature data, and no change of precision occurs in the code. Geolocation (latitude and longitude) computations are done in 32-bit and 64-bit floating point precision. The final latitude and longitude values are accurate to one meter. All time computations are done in 64-bit floating point and 64-bit integer precision.

In order to speed GTM Imagery processing, an interpolation scheme is used to calculate grid data for the granule. The latitude and longitude and associated grid row and column values are calculated for every 10^{th} point in the GTM grid. Then quadratic interpolation is performed over 20x20 pixel rectangles within the GTM grid to obtain the full geolocation for the granule. The maximum error induced by the interpolation is one meter. Row times are calculated by performing linear interpolation between the granule boundary times.

2.3.7 Algorithm Support Considerations

DMS and INF must be running before execution of the GTM Imagery algorithm.

Currently, the algorithm processes six (6) of the sixteen (16) Moderate Band (M-Band) inputs resulting in the production of six output M-Band EDRs. The selection of which six bands to create is determined by the corresponding M-Band input entries in the algorithm configuration guide XML file. The default bands are M1, M4, M9, M14, M15, and M16. It is the responsibility of the operator to select the appropriate bands for the mode of the granule (day, mixed, or night) that is to be processed.

All sixteen Moderate Band (M-Band) inputs must appear in the "Inputs" section of the configuration guide, whether or not they're actually used during processing, therefore do not remove any of them as automated source code generation phases of the software build may be affected. An M-Band input having the configuration guide Mode keyword set to "Required" will be selected for processing. If the input mode is set to "NOAUTOSETUP", the band will not be processed. Do not specify the input mode as "Optional" for any of the M-Band inputs. All M-Band inputs must specify a "Wait" keyword value of TRUE in order to process cross-granules.

Algorithm code expects exactly six of the M-Band inputs to be used and therefore exactly six must be "Required". You may change which six of the M-Band inputs are used by modifying the input Mode keyword values appropriately (a full recompile of the algorithm is required but no code changes are necessary). In the future, if the number of M-Band inputs to be processed, and resulting output EDRs, must be changed to some value other than six (6), algorithm code changes will be necessary.

The configuration file is in XML group-name / configuration entry format. Each configuration entry consists of a name / configValue pair. There are many other keys in the file, but only the "Mode" name field (Input Mode) in the configuration entries in Table 33 below should be modified. Table 33 also shows, for the default M-Band inputs named previously, the corresponding output product EDR short names.

The name of the XML configuration file is:
/vobs/PRO/cfg/<(NPP/J01)>ProEdrViirsMChannelImagery_CFG.xml.

Table 26: VIIRS M-Band Imagery Configuration

Input Group Name	Input Short Name	Input Mode	Band Processed	Output EDR
Moderate_Band01	VIIRS-M1-FSDR	Required	Yes	VIIRS-M1ST-EDR
Moderate_Band02	VIIRS-M2-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band03	VIIRS-M3-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band04	VIIRS-M4-FSDR	Required	Yes	VIIRS-M2ND-EDR
Moderate_Band05	VIIRS-M5-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band06	VIIRS-M6-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band07	VIIRS-M7-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band08	VIIRS-M8-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band09	VIIRS-M9-FSDR	Required	Yes	VIIRS-M3RD-EDR
Moderate_Band10	VIIRS-M10-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band11	VIIRS-M11-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band12	VIIRS-M12-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band13	VIIRS-M13-FSDR	NOAUTOSETUP	No	N/A
Moderate_Band14	VIIRS-M14-FSDR	Required	Yes	VIIRS-M4TH-EDR
Moderate_Band15	VIIRS-M15-FSDR	Required	Yes	VIIRS-M5TH-EDR
Moderate_Band16	VIIRS-M16-FSDR	Required	Yes	VIIRS-M6TH-EDR

2.3.8 Assumptions and Limitations

No assumptions or limitations have been identified.

3.0 GLOSSARY/ACRONYM LIST

3.1 Glossary

Table 27 contains terms most applicable for this OAD.

Table 27: Glossary

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: 1.0 A theoretical description (i.e., science/mathematical basis) 2.0 A computer implementation description (i.e., method of solution) 3.0 A computer implementation (i.e., code)
Algorithm Configuration Control Board (ACCB)	Interdisciplinary team of scientific and engineering personnel responsible for the approval and disposition of algorithm acceptance, verification, development and testing transitions. Chaired by the Algorithm Implementation Process Lead, members include representatives from IWPTB, Systems Engineering & Integration IPT, System Test IPT, and IDPS IPT.
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 IWPTB facility. Delivered code is executed on compatible IWPTB 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.
cm	Centimeter - unit of measurement for length.
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 ATBD. At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Environmental Data Record (EDR)	*[IORD Definition]* 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.). *[Supplementary Definition]* 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 JPSS SDRs or EDRs, plus ancillary environmental data provided by others. EDR metadata contains references to its processing history, spatial and temporal coverage, and quality.
IDPS Epoch Time (IET)	The standard for IDPS time storage. IET is the actual elapsed microsceonds, on the International Atomic Clock, based on an epoch date of 01 Jan 1958 (start of the International Geophysical Year. Also the base for TAI time)
K	Kelvin - unit of measurement for temperature.
M/s	Meters per second - unit of measurement for velocity.
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)	*[IORD Definition]* Full resolution digital sensor data, time referenced, 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. *[Supplementary Definition]* 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.
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 ATBD. 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 Algorithm Theoretical Basis Document (ATBD). At a minimum, implemented software is science-grade and includes test data demonstrating data quality compliance.
Sensor Data Record (SDR)	*[IORD Definition]* Data record produced when an algorithm is used to convert Raw Data Records (RDRs) to calibrated brightness temperatures with associated ephemeris data. 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. The existence of the SDRs provides reversible data tracking back from the EDRs to the Raw data. *[Supplementary Definition]* 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.
Tau	Unit of measurement for Optical Thickness.

Term	Description
Temperature Data Record (TDR)	*[IORD Definition]* 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. *[Supplementary Definition]* 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.
ViirsAmiIPType	VIIRS Aerosol Model Index Intermediate Product. Data is stored in an array of unsigned 8-bit integers.
ViirsAotIPType	VIIRS Aerosol Optical Thickness Intermediate Product. Data is stored in an array of 32-bit floating point numbers.
ViirsCloudMask IPType	A 48-bit word (6 bytes) for each moderate resolution pixel that includes information about whether the view of the surface is obstructed by clouds and specifies the processing path the algorithm took. Cloud phase data is also included as well as spatial uniformity, aerosol, shadow, and fire detection data.
ViirsIceConcIP Type	VIIRS Ice Concentration Intermediate Product. Data is stored in an array of 32-bit floating point numbers.
ViirsModBtType	VIIRS Moderate Resolution Channel Brightness Temperature. Data is stored in an array of 32-bit floating point numbers.
ViirsPwIPType	VIIRS Precipitable Water Intermediate Product. Data is stored in an array of unsigned 8-bit integers.
ViirsSnowIce CoverIPType	VIIRS Snow Ice Cover Intermediate Product. Data is stored in an array of 32-bit floating point numbers.
ViirsSelectSstLut Type	VIIRS Select SST Look Up Table.
ViirsSstCoeffsLut Type	VIIRS SST Coefficient Look Up Table.

3.2 Acronyms

Table 28 contains terms most applicable for this OAD.

Table 28: Acronyms

Acronym	Description
AM&S	Algorithms, Models & Simulations
API	Application Programming Interfaces
ARP	Application Related Product
DMS	Data Management Subsystem
DQTT	Data Quality Test Table
IET	IDPS Epoch Time
IIS	Intelligence and Information Systems
INF	Infrastructure
ING	Ingest
IP	Intermediate Product
LUT	Look-Up Table
MDS	Map Data Set
NCC	Near Constant Contrast
PDL	Program Design Language
PRO	Processing
QF	Quality Flag
SDR	Sensor Data Records
SI	Software Item or International System of Units
SOM	Space Oblique Mercator (a conformal nearly equal area map)
TBD	To Be Determined
TBR	To Be Resolved
TOA	Top of the Atmosphere

4.0 OPEN ISSUES

Table 29: List of OAD TBD/TBR

No.	Description	Resolution Date
None		