Effective Date: August 09, 2023
Expiration Date: August 09, 2028

GSFC JPSS CMO
November 14, 2023
Released

474-00448-03-26, Revision A
Joint Polar Satellite System (JPSS) Code 474

Joint Polar Satellite System Algorithm Specification Volume III: Operational Algorithm Description (OAD) for the VIIRS Imagery

NOAA / NASA

Goddard Space Flight Center
Greenbelt, Maryland

Joint Polar Satellite System
Algorithm Specification Volume III:
Operational Algorithm Description (OAD) for the VIIRS Imagery
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

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

Revision	Effective Date	Description of Changes (Reference the CCR & CCB/ERB Approval Date)
Rev -	Aug 29, 2013	This was approved by the Ground ERB via **474-CCR-13-1178** on the effective date shown.
Rev -1	Oct 23, 2014	This version incorporates **474-CCR-14-2094** 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-19-4790, 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

List of Figures

List of Tables

1 INTRODUCTION

1.1 Objective

The purpose 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 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.

1.2 Scope

The scope of this document covers two OADs.

First is the OAD description of the core operational algorithm(s) required to create the VIIRS Near Constant Contrast (NCC) EDR.

The second OAD includes 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.

2 RELATED DOCUMENTATION

The latest JPSS document(s) can be obtained from URL: 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
474-00448-01-26	JPSS Algorithm Specification Volume I: SRS for the VIIRS Imagery

2.2 Applicable Documents

The following document is the Applicable Document 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
474-00448-02-26	JPSS Algorithm Specification Volume II: Data Dictionary for the VIIRS Imagery

3 VIIRS NCC ALGORITHM OVERVIEW

This section describes the operational algorithm that produces VIIRS sensor NCC Imagery EDR. The NCC product is created from the Day/Night Band (DNB) SDR, where the DNB data is mapped to the Coarse GTM map and then processed in such a way to minimize the apparent transition in radiance across the terminator. For more information on the specifics of the GTM mapping see the Imagery section 4.4; and the Operational Algorithm Description Document for VIIRS Ground Track Mercator (GTM) Imagery Environmental Data Record (EDR) Software. The VIIRS NCC Imagery EDR is computed after the SDR process is complete. Figure 3-1 illustrates this processing relationship.

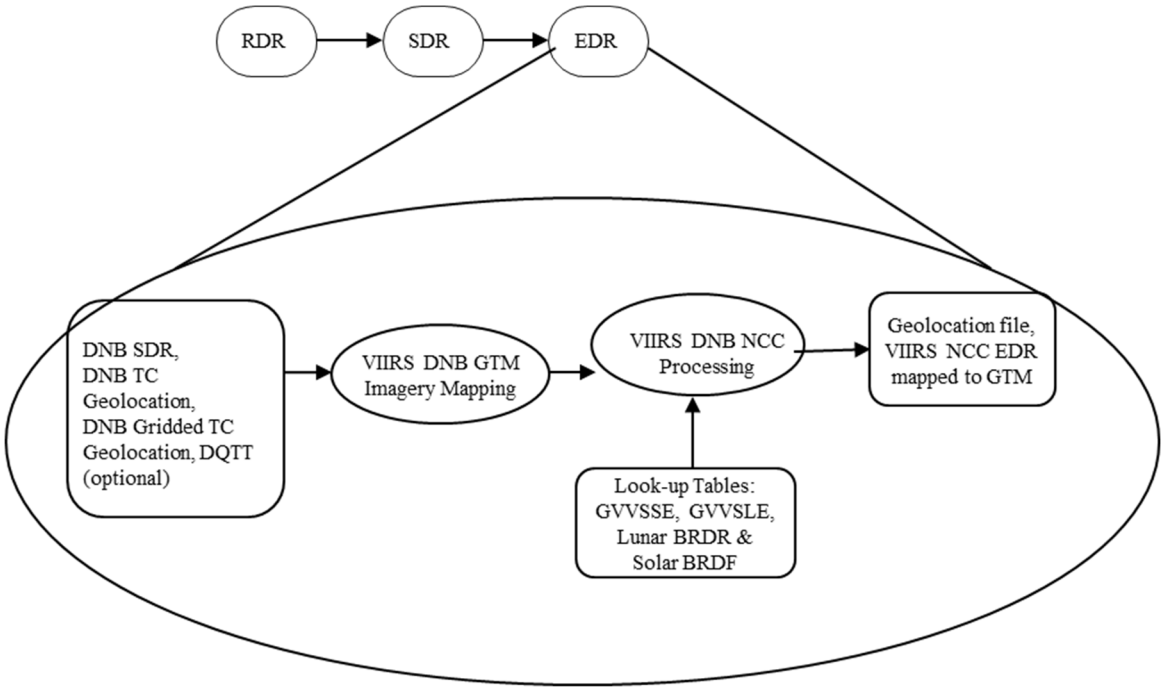


Figure 3-1. Processing Chain Associated with VIIRS NCC Imagery EDR

NCC Visible Imagery is derived from the DNB measured in regions with solar illumination in daytime, with lunar illumination at night, or near the terminator (twilight) region. Due to the significant dynamic range of solar and lunar irradiance on the earth, a three-stage Charge Coupled Device (CCD) sensor is designed to record the radiance in this region. The individual detectors in the three stages of the CCD are used to record daytime, twilight and nighttime radiance. The SDR for this band consists of a single set of radiance data containing all three stages merged to include all illumination conditions. DNB calibration provides calibrated Top of Atmosphere (TOA) radiance over a dynamic range of 3×10^{-9} to 2×10^{-2} **W• cm^{-2}•sr^{-1}**. The NCC algorithm converts the TOA radiance to an imagery product by removing variation due to the solar and lunar source irradiance for each pixel. The algorithm preserves heritage from the Operational Linescan System (OLS) Gain Management Algorithm (GMA) using Look-Up Tables (LUTs) designed to mimic the GMA.

3.1 VIIRS NCC Imagery Description

3.1.1 Interfaces

The VIIRS NCC Imagery algorithm is initiated by an IDPS Infrastructure (INF) subsystem Software Item (SI) to process the data. This INF SI provides tasking information to the algorithm indicating which granule to process. The Data Management Subsystem (DMS) SI provides data storage and retrieval capability. A library of C++ classes is used to implement SI interfaces. More information regarding these topics is found in document UG60917-IDP-1005 with reference in particular to sections regarding PRO Common (CMN) processing and the IPO Model.

3.1.1.1 Inputs

Computing the VIIRS NCC Imagery EDR requires several types of data.

Table 3.1.1.1-1. NCC EDR Inputs

Input	Description	Reference Documents
VIIRS DNB SDR	VIIRS Calibrated TOA Radiances for DNB.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS DNB Terrain Corrected Geolocation File	Terrain Corrected Earth location for each satellite view point as well as solar, lunar, and view geometry.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS Day Night Band Terrain Corrected Gridded Geolocation	Terrain Corrected Grid row and column values for every pixel in the granule and the granule MDS.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
Solar GVVSE Look-Up Table (VIIRS-Ga-Val-Vs-Scene-Sol-Elev-LUT)	LUT containing gain values for the solar signal as a function of solar zenith angle in the DNB.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
Solar BRDF Look-Up Table (VIIRS-Sol-BRDF-LUT)	LUT providing the anisotropic reflectance factors for the solar signal in the DNB as a function of the illumination angles and the sensor zenith angles. Note that this is not strictly speaking a true BRDF, because it does not contain units of inverse steradians. The actual BRDF can be determined from this LUT by dividing by pi.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
Lunar GVVSE Look-Up Table (VIIRS-Ga-Val-Vs-Scene-Lun-Elev-LUT)	LUT containing gain values for the lunar signal	474-00448-02-26_JPSS-DD-Vol-II-Part-26

Input	Description	Reference Documents
	as a function of lunar zenith angle in the DNB.	
Lunar BRDF Look-Up Table (VIIRS-Lun-BRDF-LUT)	LUT providing the anisotropic reflectance factors for the lunar signal in the DNB as a function of the illumination angles and the sensor zenith angles. Note that this is not strictly speaking a true BRDF, because it does not contain units of inverse steradians. The actual BRDF can be determined from this LUT by dividing by pi.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
Lunar Phase Look-up Table (VIIRS-LUN-Phase-LUT)	LUT providing the lunar radiance as a function of lunar phase angles.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
Threshold Look-up Table	LUT providing thresholds for the algorithm.	474-00448-02-26_JPSS-DD-Vol-II-Part-26
Data Quality Threshold Table	Reports erroneous pixels through a DQN. Performs a bitmask tests on quality flag bits 0-1 for red condition (NCC Imagery quality poor).	474-00448-02-26_JPSS-DD-Vol-II-Part-26

3.1.1.2 Outputs

The VIIRS NCC Imagery algorithm produces a data item containing calculated NCC Imagery values and a data item containing geolocation data.

Table 3.1.1.2-1. NCC EDR Outputs

Outputs	Description	Reference Document
VIIRS NCC EDR	VIIRS-NCC-EDR contains [scaled] data fields, unscaled EDR products and Quality flags	474-00448-02-26_JPSS-DD-Vol-II-Part-26 474-00448-01-26_JPSS-SRS-Vol-I-Part-26
VIIRS NCC Geolocation	**NCC EDR Geolocation Data**	474-00448-02-26_JPSS-DD-Vol-II-Part-26 474-00448-01-26_JPSS-SRS-Vol-I-Part-26

3.1.2 NCC Algorithm Processing

The purpose of the VIIRS NCC Imagery Unit is to derive NCC Imagery for each pixel of the DNB Visible imagery (mapped to the Coarse GTM map) and to write the VIIRS NCC Imagery EDR. The NCC processing specific source code is written in FORTRAN 90 with the interface to

IDPS written in C++. Each of the following routines is presented with a brief description of their function.

NCC Imagery is derived for each DNB pixel, using the VIIRS DNB SDR and LUTs as input data files. The class responsible for retrieving data from and putting data into DMS is contained in the file ProEdrViirsGtmNccImagery.cpp.

The program IM_main.f contains all NCC processing specific subroutines. Figure 3.1.2-1 depicts a Level 2 data flow diagram for the VIIRS NCC Imagery EDR and Figure 3.1.2-2 illustrates the process of how NCC Imagery is produced from the DNB SDR. Here both Moon and Sun DNB scene illuminations are identified with computation of radiance from each source. These radiance contributions are combined into a total source radiance to compute the NCC Imagery product.

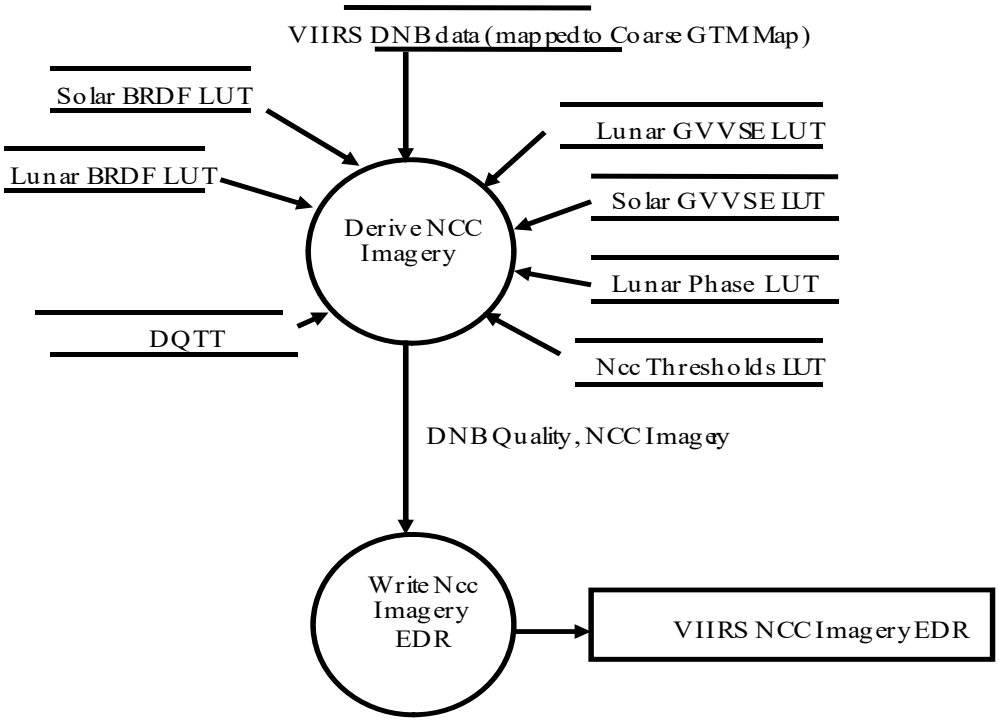


Figure 3.1.2-1. Generate NCC VIIRS Imagery EDR Level 2 Data Flow Diagram

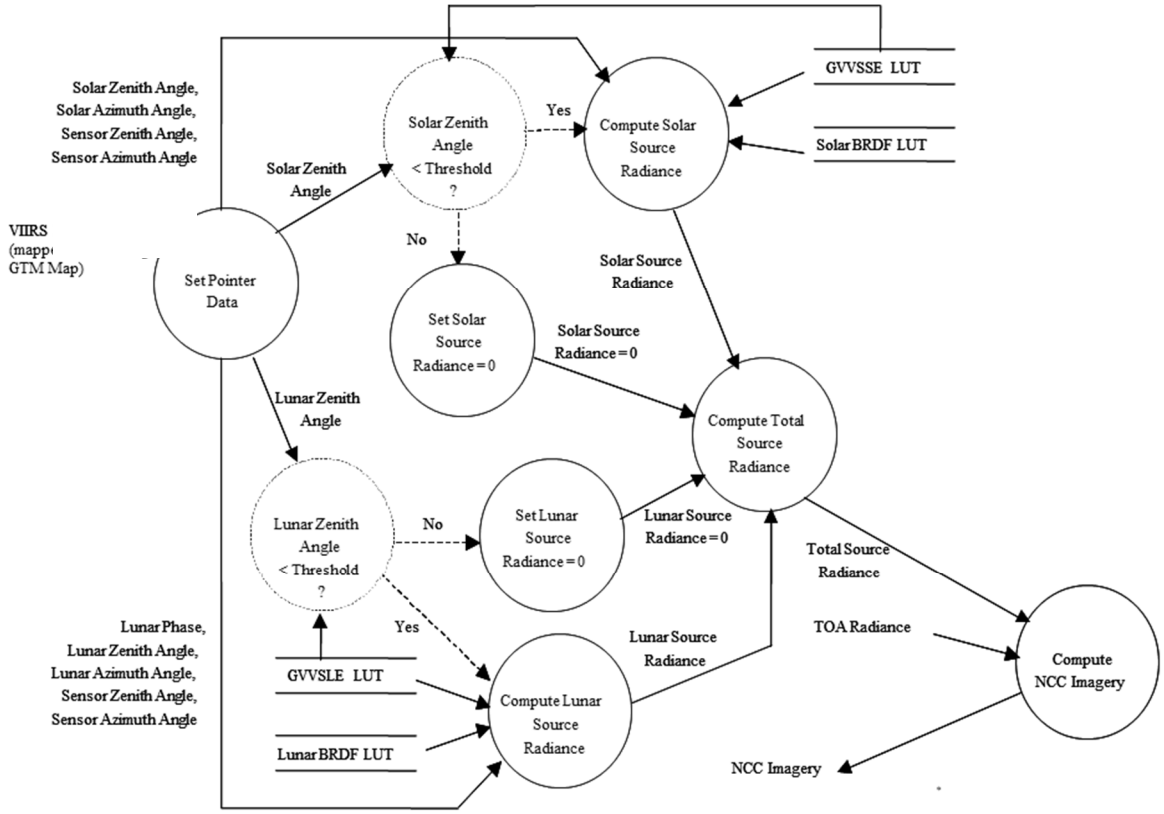


Figure 3.1.2-2. Derived NCC Imagery Level 3 Data Flow Diagram

ProEdrViirsGtmNccImagery is the derived algorithm for the NCC Imagery algorithm. It is a subclass of the ProEdrViirsGtmImagery class which is, in turn, a subclass of the ProCmnAlgorithm class. This derived algorithm class creates a list of input data items read from DMS, performs the GTM mapping, and passes required data into the NCC algorithm. An output data item is written to DMS once both of the algorithms finish processing this data. The GTM algorithm calculates the GTM geolocation and mapping information by calling the createGTM() method. Next, this class copies the DNB SDR radiance data into a temporary buffer and passes the temporary buffer into the NCC Imagery IM_main routine. All further references to "DNB pixel" or "DNB scene" in this document are referring to this temporary buffer of DNB radiances which has been mapped to the GTM space. See the VIIRS Ground Track Mercator Imagery EDR OAD for more information.

3.1.2.1 Main Module – IM_main

This routine is the main driver for the NCC Imagery EDR. This program generates NCC Imagery from DNB data using OLS heritage, current atmospheric data, solar, lunar and sensor geometry. The solar and lunar radiances are computed for each DNB pixel.

3.1.2.1.1 Compute Lunar Gain (IM_lunar)

This subroutine calculates Lunar Gain and Lunar BRDF at each DNB pixel. The Lunar Gain LUT is interpolated in lunar zenith angle to estimate the gain factor for every pixel in the DNB scene. The Lunar BRDF LUT is interpolated on lunar phase, lunar zenith angle, sensor zenith

angle, and relative azimuth to estimate the value of the anisotropic reflectance factor for every pixel in the DNB scene.

3.1.2.1.2 Compute Lunar Gain (IM_lunar)

This subroutine calculates Lunar Gain and Lunar BRDF at each DNB pixel. The Lunar Gain LUT is interpolated in lunar zenith angle to estimate the gain factor for every pixel in the DNB scene. The Lunar BRDF LUT is interpolated on lunar phase, lunar zenith angle, sensor zenith angle, and relative azimuth to estimate the value of the anisotropic reflectance factor for every pixel in the DNB scene.

3.1.3 Graceful Degradation

None.

3.1.4 Exception Handling

The VIIRS NCC Imagery Unit software is designed to handle a wide variety of processing problems, including bad and missing data and fatal errors. Any exceptions or errors are reported to IDPS using the appropriate INF Application Program Interface (API). Three possible quality flags (QF) have placeholders in the code for RED (3), GREEN (0) and YELLOW (1), but the YELLOW flags are not used in the current version. Since QFs are passed through from the SDR, the set of possible QFs is not limited to these three, but is determined by the SDR definition.

Error flag information is written as a QF in the event that processing problems prevent production of useful EDR data for some pixels. The NCC QF (NCC_Qual_Dnb in), however, reflects both the quality of the DNB SDR, as well as the quality of the NCC process. When the NCC algorithm encounters DNB pixels with a green QF from the SDR, but with either the minimum solar zenith angle or minimum lunar zenith angle greater than configurable thresholds, it sets the NCC_QF to RED. The current implementation altered these angle limits to allow computations under all possible and realistic values for the solar/lunar zenith angles (0 - 180 degrees), therefore provided the DNB SDR is green and there is no error in these angles, the NCC_QF will also be green.

Since the NCC outputs a floating-point number, slightly negative outputs are possible and could, in fact, result from a noisy pixel with very low radiance. The NCC Imagery output does allow for small negative radiances via a tunable threshold. If the DNB radiance falls below that threshold, a fill value for floating point real values is set to -999.9 to indicate that a value was not computed.

3.1.5 Data Quality Monitoring

Each algorithm uses specific criteria contained in a Data Quality Threshold Table (DQTT) to determine when a Data Quality Notification (DQN) is produced. The DQTT contains the thresholds used to trigger DQNs as well as the text contained in the DQN. If a threshold is met, the algorithm stores a DQN in DMS indicating the test(s) that failed and the value of the DQN attribute. For more algorithm specific detail refer to the 474-00448-01-26_JPSS-SRS-Vol-I-Part-26.

3.1.6 Computational Precision Requirements

The NCC algorithm does computations in 32-bit precision float. Despite the large dynamic range of the DNB SDR, double precision is not required. A 32-bit float value is more than sufficient to represent the seven orders of magnitude dynamic range in the radiances.

Optimization for the Solar and Lunar gain factor is based upon angle values measured in degrees. DNB geolocation angle values input by this process are already in degrees and no conversion from radians is necessary.

3.1.7 Algorithm Support Considerations

Any thresholds used in the algorithm that can be changed on a frequent basis (i.e., referred to as settable parameters) are contained within a DMS algorithm specific thresholds file. The INF and DMS must be running before the algorithm is executed.

3.1.8 Assumptions and Limitations

None.

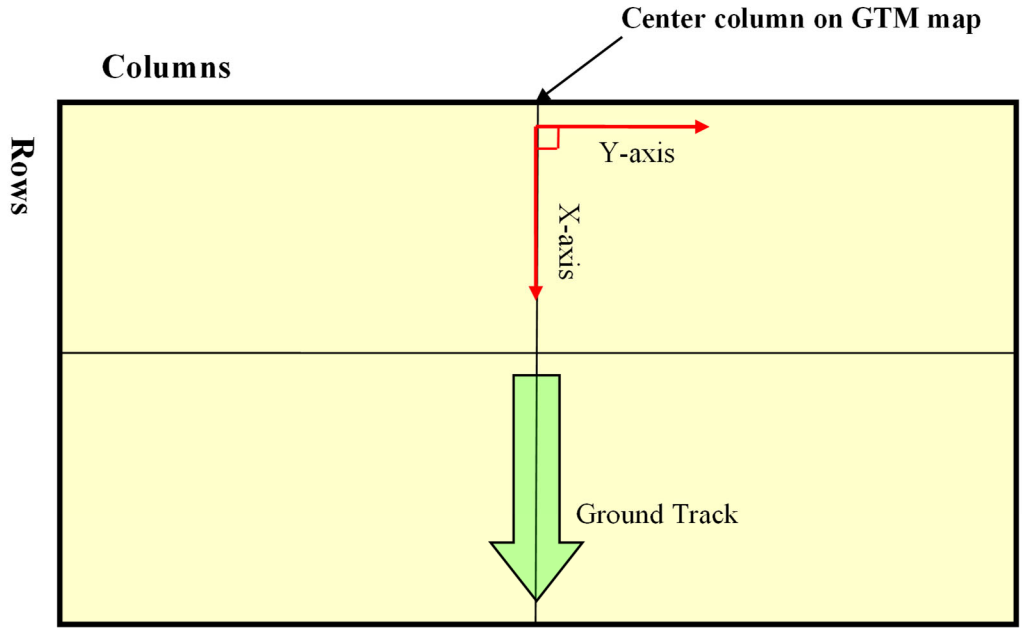
4 VIIRS GTM EDR ALGORITHM REFERENCES

The purpose of the VIIRS GTM Imagery algorithm is to map VIIRS Imaging (I) channel, Moderate (M) channel, and Day Night Band (DNB) 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.

4.1 VIIRS GTM EDR 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 4.1-1: GTM Map Attributes.



GTM Map Plan

Figure 4.1-1. 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 4.1-2. 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.

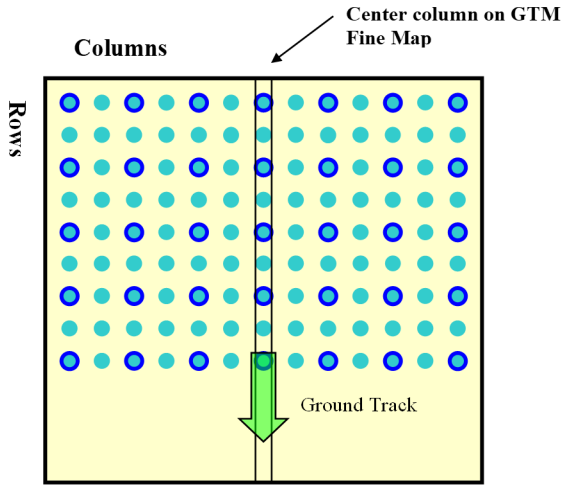


Figure 4.1-2. Fine Map Pixels with Emphasized Coarse Pixels

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.

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.

The operational software only does full calculations for every 10^{th} 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. All 16 moderate resolution channels are mapped onto the GTM map. The Day Night Band (DNB) is processed by the Near Constant Contrast (NCC) Imagery algorithm to produce an EDR mapped to GTM.

4.1.1 Interfaces

4.1.1.1 Inputs

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

4.1.1.2 Outputs

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

4.1.2 VIIRS GTM Imagery Algorithm Processing

4.1.2.1 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.

4.1.2.1.1 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.

4.1.2.1.2 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.

4.1.2.1.3 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.

4.1.2.2 Derived Algorithms

4.1.2.2.1 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

4.1.2.2.2 fillOutputStructures in 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.

4.1.2.2.3 calculateSdrPixelLocations in 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.

4.1.2.3 Reuse C Functions

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.
    1. Invoke gtm_grndtrk_ptr() to find the nadir point data for the end of the granule.
    2. Invoke gtm_grndtrk_ptr() to find the begin track point from begin time and gtm_ephem.
    3. Calculate the distance from granule begin nadir point to granule end nadir point. The earth radius is derived for the average latitude.
    4. Adjust the distance between rows so that a GTM row nadir point will be exactly on the granule end nadir point.
    5. 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.
    6. 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 4.1.2.3-1. 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.

4.1.2.3.1 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 4.1.2.3.1-1. 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

4.1.2.3.2 Bld_full_img_gtme

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 4.1.2.3.2-1. 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

4.1.2.3.3 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 4.1.2.3.3-1. 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.

4.1.2.3.4 Short_dist

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

Table 4.1.2.3.4-1. 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

4.1.2.3.5 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 44.1.2.3.5-1 shows the azm_sidb parameter definitions.

Table 4.1.2.3.5-1. 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.

4.1.2.3.6 Target_pt

Below is a discussion of function target_pt, which does all of the spherical trig calculations described in Section 3.0, Algorithm Overview.

Figure 4.1.2.3.6-1. 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.

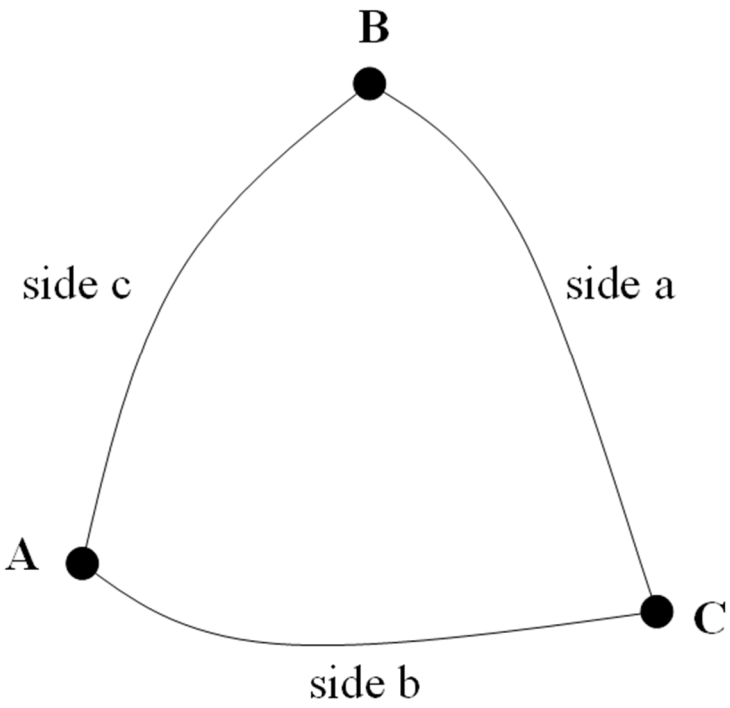


Figure 4.1.2.3.6-1. Target_pt Function Calculations Diagram

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 4.1.2.3.6-1 shows the target_pt parameter definitions.

Table 4.1.2.3.6-1. 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).

4.1.2.3.7 Grid_to_img_Sdrpixel

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 4.1.2.3.7-1 shows the grid_to_imgSDRpixel parameter definitions.

Table 4.1.2.3.7-1. 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.

4.1.2.3.8 Grid_to_mod_Sdrpixel

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 4.1.2.3.8-1 shows the grid_to_modSDRpixel parameter definitions.

Table 4.1.2.3.8-1. 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.

4.1.2.3.9 Grid_to_dnb_Sdrpixel

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 4.1.2.3.9-1 shows the grid_to_imgSDRpixel parameter definitions.

Table 4.1.2.3.9-1. 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.

4.1.2.3.10 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 4.1.2.3.10-1 shows the rp_g2imgpix parameter definitions.

Table 4.1.2.3.10-1. 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.
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.

4.1.2.3.11 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. **Error! Reference source not found.**4.1.2.3.11-1 shows the rp_g2modpix parameter definitions.

Table 4.1.2.3.11-1. 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.

4.1.2.3.12 Rp_g2dnbpix

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 4.1.2.3.12-1 shows the rp_g2dnbdpix parameter definitions.

Table 4.1.2.3.12-1. 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.

4.1.2.3.13 Grid_to_latlon

This function converts a row column position from an MDS to a latitude and longitude. Table 4.1.2.3.13-1 shows the grid_to_latlon parameter definitions.

Table 4.1.2.3.13-1. 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.

Parameter	Type	I/O	Description
Err	Int	O	Returned error code.

4.1.2.3.14 Latlon_to_grid

This function converts a latitude and longitude to a position on a Map Data Set grid. Table 4.1.2.3.14-1 shows the latlon_to_grid parameter definitions.

Table 4.1.2.3.14-1. 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.

4.1.2.3.15 Earth_radius_D

This function computes radius of the Earth, in kilometers, from the geodetic latitude. Table 4.1.2.3.15-1 shows the earth_radius_D parameter definitions.

Table 4.1.2.3.15-1. 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.

4.1.2.3.16 Gridtoimg_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 4.1.2.3.16-1. grid2img_pix_wiscan Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	interpolation grid row
gcol	Double	I	interpolation grid column
vI_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

4.1.2.3.17 Gridtomod_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 4.1.2.3.17-1. grid2mod_pix_wiscan Parameter Definitions

Parameter	Type	I/O	Description
grow	Double	I	interpolation grid row
gcol	Double	I	interpolation grid column

Parameter	Type	I/O	Description
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

4.1.2.3.18 gridtodnb_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 4.1.2.3.18-1. 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

4.1.3 Graceful Degradation

None.

4.1.4 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.

4.1.5 Data Quality Monitoring

None.

4.1.6 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 10th 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.

4.1.7 Algorithm Support Considerations

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

4.1.8 Assumptions and Limitations

None.

4.2 VIIRS 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 4.2-1. Basic Processing Flow for the VIIRS I-Band Imagery EDR Inputs are the VIIRS SDRs (channels I1 through I5), VIIRS I-Band Terrain Corrected Geolocation grid data, and VIIRS I-Band sensor look angles. Outputs are the I-Band Imagery EDRs and geolocation data.

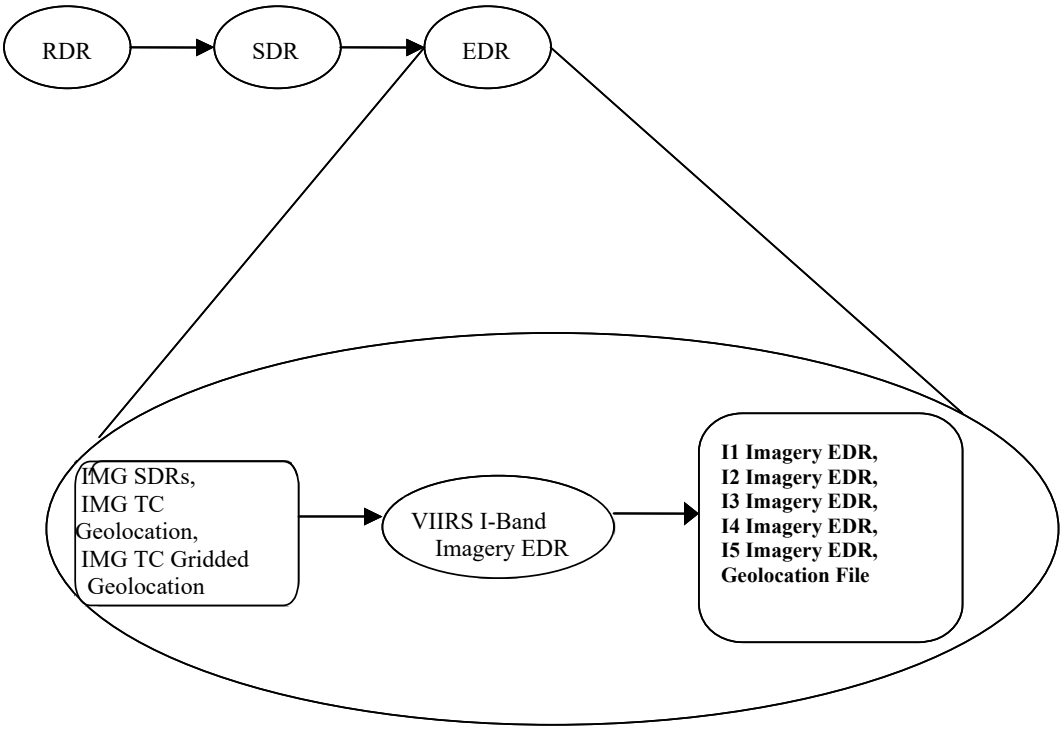


Figure 4.2-1. Basic Processing Flow for the VIIRS I-Band Imagery EDR

4.2.1 Interfaces VIIRS GTM Imagery Base Algorithm Description

4.2.2 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 4.2.2-1. 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 TC Geolocation	Terrain Corrected 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 TC Gridded Geolocation	Terrain Corrected 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

4.2.3 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 4.2.3-1. 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

4.2.4 Algorithm Processing

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.

4.2.4.1 Main Module -ProEdrViirsGtmBandImagery.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.

4.2.4.1.1 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.

4.2.4.1.2 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.

4.2.4.1.3 InitOutputDataItems

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

4.3 Graceful Degradation

None.

4.4 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.

4.5 Data Quality Monitoring

None.

4.6 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.

4.7 Algorithm Support Considerations

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

4.8 Assumptions and Limitations

None.

5 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 5-1. Inputs are the VIIRS Mod SDRs, VIIRS Mod Terrain Corrected Geolocation grid data, and Mod VIIRS sensor look angles. Outputs are the M-Band Imagery EDRs and geolocation data.

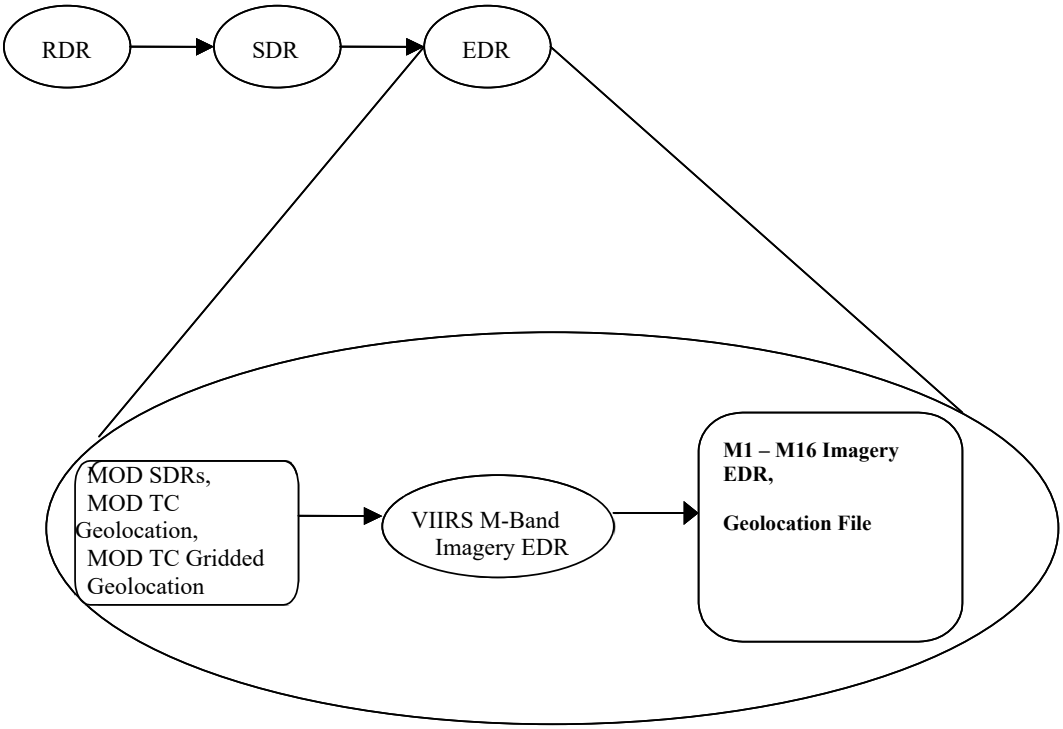


Figure 5-1. Basic Processing Flow for the VIIRS M-Band Imagery EDR

5.1 Interfaces

5.1.1 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 5.1.1-1. 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 TC Geolocation	Terrain Corrected 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 TC Gridded Geolocation	Terrain Corrected 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

5.1.2 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 5.1.2-1. VIIRS M-Band Imagery EDR Outputs

Name	Description	Reference Documents
VIIRS Band M1 – M16 Moderate Band EDRs	All M Band EDRs except for M13 contain scaled radiance/reflectance/brightness temperature. M13 EDR is unscaled.	474-00448-01-26_JPSS-SRS-Vol-I-Part-26 474-00448-02-26_JPSS-DD-Vol-II-Part-26
VIIRS Moderate Band 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

5.2 Algorithm Processing

5.2.1 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. One geolocation dataset is also created per granule.

5.2.2 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.

5.2.3 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.

5.2.4 initOutputDataItems

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

5.3 Graceful Degradation

None.

5.4 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.

5.5 Data Quality Monitoring

None.

5.6 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.

5.7 Algorithm Support Considerations

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

5.8 Assumptions and Limitations

No assumptions or limitations have been identified.

6 GTM IMAGERY NCC CLASS DESCRIPTION

ProEdrViirsGtmNccImagery is the main GTM NCC Imagery process. It instantiates an NCC instance of the ProEdrViirsGtmImagery template class, by deriving from the auto-generated NCC algorithm class, “AutoGeneratedProEdrViirsGtmNccImagery”, and calls the inherited applyAlgorithm() method.

The GTM mapping of the DNB SDR data to the GTM layout are detailed above in 4.2. **Error! Reference source not found.**. Inputs are the VIIRS DNB SDR, VIIRS DNB Terrain Corrected grid data, and DNB VIIRS sensor look angles. Outputs are the NCC Imagery EDRs and geolocation data.

7 GLOSSARY/ACRONYM LIST

7.1 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: 1. A theoretical description (i.e., science/mathematical basis) 2. A computer implementation description (i.e., method of solution) 3. A computer implementation (i.e., code).
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 JPSS System, but which is acquired from external providers and used by the JPSS system in the production of JPSS data products.
Auxiliary Data	Auxiliary Data is defined as data, other than data included in the sensor application packets, which is produced internally by the JPSS system, and used to produce the JPSS deliverable 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)	*[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 microsceconds, 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)
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

Term	Description
	(APIs) as specified for TDR/SDR or EDR code.
Raw Data Record (RDR)	*[IORD Definition]* 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. *[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 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 Algorithm Document (OAD). 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. 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.
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

Term	Description
	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.

7.2 Acronyms

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

Acronym	Description
AM&S	Algorithms, Models & Simulations
API	Application Programming Interfaces
CMN GEO	Common Geolocation
CSN	Collection Short Name
DMS	Data Management Subsystem
DQTT	Data Quality Test Table
E&A	Ephemeris and Attitude
ECEF	Earth-Centered Earth-Fixed
ECF	Earth-Centered Fixed
ECR	Earth-Centered Rotating
ECSF	Earth-Centered Space-Fixed
EES	Ellipsoid to Earth's Surface
EGM	Earth Geoid Model
EGS	Ellipsoid Geoid Separation
EPHATT	Ephemeris ad Attitude Structure
IET	IDPS Epoch Time is the time in number of microseconds since 1-1-1958 00:00:00
IIS	Intelligence and Information Systems
INF	Infrastructure
ING	Ingest
IP	Intermediate Product
LOS	Line of Sight
LUT	Look-Up Table
MDS	Map Data Set
MSL	Mean Sea Level
NOVAS-C	Naval Observatory Vector Astrometry System – C version
NCC	Near Constant Contrast
PRO	Processing
QF	Quality Flag
SAA	South Atlantic Anomaly
SDR	Sensor Data Records
SI	Software Item or International System of Units
SOM	Space Oblique Mercator (a conformal nearly equal area map)
STL	Standard Template Library
TBD	To Be Determined
TBR	To Be Resolved
TLE	Two Line Element files following the format specified by NORAD
TOA	Top of the Atmosphere
UT1	Universal Time One

Acronym	Description
UTC	Universal Time Coordinated