Effective Date: April 22, 2011
Revision -

GSFC JPSS CMO
08/06/2011
Released

Joint Polar Satellite System (JPSS) Ground Project
Code 474
474-00044

Joint Polar Satellite System (JPSS) VIIRS Cloud Cover/Layers Algorithm Theoretical Basis Document (ATBD)

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.

Goddard Space Flight Center
Greenbelt, Maryland

National Aeronautics and
Space Administration

Check the JPSS MIS Server at https://jpssmis.gsfc.nasa.gov/frontmenu_dsp.cfm to verify that this is the correct version prior to use.

This page intentionally left blank.

Joint Polar Satellite System (JPSS)
VIIRS Cloud Cover/Layers
Algorithm Theoretical Basis Document (ATBD)

JPSS Electronic Signature Page

Prepared By:

Neal Baker
JPSS Data Products and Algorithms, Senior Engineering Advisor
(Electronic Approvals available online at https://jpssmis.gsfc.nasa.gov/mainmenu_dsp.cfm)

Approved By:

Heather Kilcoyne
DPA Manager
(Electronic Approvals available online at https://jpssmis.gsfc.nasa.gov/mainmenu_dsp.cfm)

Goddard Space Flight Center
Greenbelt, Maryland

This page intentionally left blank.

Preface

This document is under JPSS Algorithm 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 Ground Project Configuration Management Office
NASA/GSFC
Code 474
Greenbelt, MD 20771

This page intentionally left blank.

Change History Log

Revision	**Effective Date**	**Description of Changes** (Reference the CCR & CCB/ERB Approve Date)
Original	04/22/2011	**474-CCR-11-0059**: This version baselines D43317, VIIRS Cloud Cover/Layers Algorithm Theoretical Basis Document (ATBD), Rev B dated 11/28/2007 as a JPSS document, version Rev –. This is the version that was approved for NPP launch. Per NPOESS CDFCB - External, Volume V – Metadata, doc number D34862-05, this has been approved for Public Release into CLASS. This CCR was approved by the JPSS Algorithm ERB on April 22, 2011.

This page intentionally left blank.

D43317

Rev B

NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

VIIRS CLOUD COVER/LAYERS

Algorithm Theoretical Basis Document (ATBD)

CDRL No. A032

Northrop Grumman Space & Mission Systems Corporation
One Space Park
Redondo Beach, California 90278

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.

The information provided herein does not contain technical data as defined in the International Traffic in Arms Regulations (ITAR) 22 CFR 120.10.

This document has been approved by the Unites States Government for public release in accordance with NOAA NPOESS Integrated Program Office.

Distribution: Statement A: Approved for public release; distribution is unlimited.

NATIONAL POLAR-ORBITING OPERATIONAL ENVIRONMENTAL SATELLITE SYSTEM (NPOESS)

VIIRS CLOUD COVER/LAYERS

Algorithm Theoretical Basis Document (ATBD)

PREPARED BY:

Merit Shoucri

ELECTRONIC APPROVAL SIGNATURES:

____

Roy Tsugawa Date

Algorithm & Data Processing IPT Lead &

Algorithm Change Control Board Chairperson

____

Gerald J. Mulvey Date

Senior Systems Engineer

The following individuals are recognized for their contributions to the current or previous versions of this document.

Richard Slonaker, Igor Appel, H.-L. Huang, H.M. Woolf, R.L. Solomon

Revision/Change Record

Document Number D43317

Revision	Document Date	Revision/Change Description	Pages Affected
---	01/18/2007	PCIM Release to bring document into Matrix Accountability. Reference original document number: Y2392 delivered in 2003	All
A	11/28/2007	PCIM Release to bring document into Matrix Accountability. Reference original AER document number: P1187-TR-I-006 delivered in 2005	All
B	11/28/2007	In preparation for Public Release of this ATBD, the following administrative changes were made: All ITAR markings were removed, and Distribution Statement F added. Incorporates ECR A-057. Approved for Public Release per Contracts Letter 100730-1.	All

Revision/Change Record

For Document No. **P1187-TR-I-006**

Revision	Document Date	Revision/Change Description	Pages Affected
Ver 5		Release from ITSS as Document Y2392	All
Vers 6, Rev 0	5 Jan 2005	Release from AER (E. Kennelly, D. Hogan)	All
Vers 6, Rev 1	21 Jan 2005	Release from Northrop Grumman, SPCR 0722 Updates Addition of ECEF, ECR, ECI, NIMA to Glossary Addition of document numbers to VIIRS Documents section Addition of Parallax Correction Algorithm Description Grammatical and document format corrections Addition of SDP Toolkit and NIMA reference	 viii, ix 1 27 various 69
Ver 6, Rev 2beta	26 May 2005	Release from AER addressing new cloud mask and multi-layer cloud phase	13, 34
Ver 6, Rev 2	16 Jun 2005	Release from AER. Resolved discrepancies between prior revisions. Includes updates associated with new cloud mask and strategy for addressing multi-layer cloud.	13, 34

Preface to Version 6

Version 6 of this ATBD was produced by Atmospheric and Environmental Research, Inc. by revising Version 5, Release 3 of the ATBD (authored by Raytheon System Company, Information and Technology Services). This version incorporates the following algorithm changes: (1) the k-Means clustering algorithm was modified to improve performance and be more robust to missing input data; (2) the output EDR horizontal cells now have approximately constant horizontal cell size across the scan.

The authors from Raytheon and AER are listed below:

Version 6
ATMOSPHERIC AND ENVIRONMENTAL RESEARCH, INC.
131 Hartwell Avenue
Lexington, MA 02139

Ted Kennelly
David Hogan

Through Version 5 (SBRS Document #: Y2392)
RAYTHEON COMPANY
Information Technology and Scientific Services
1616 McCormick Drive
Upper Marlboro, MD 20774

Hung-Lung A. Huang
Hal M. Woolf
Ryan L. Solomon
Richard L. Slonaker

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

GLOSSARY OF ACRONYMS

ATBD	Algorithm Theoretical Basis Document
AVHRR	Advanced Very High Resolution Radiometer
CC/L	Cloud Cover/Layers
CDR	Critical Design Review
COT	Cloud Optical Thickness
CTH	Cloud Top Height
DoD	Department of Defense
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
ECR	Earth Centered Rotating
EDR	Environmental Data Record
EOS	Earth Observing System
EPS	Effective Particle Size
ETM	Enhanced Thematic Mapper (Landsat)
GIFOV	Ground Instantaneous Field of View
GSD	Ground Sample Distance
HC	Horizontal Cell
HCS	Horizontal Cell Size
HSI	Horizontal Sample Interval
HSR	Horizontal Spatial Resolution
IP	Intermediate Product
LLLS	Low Level Light Sensor
LWIR	Longwave Infrared
METEOSAT	Geosynchronous METEOrological SATellite
MODIS	Moderate Resolution Imaging Spectroradiometer
MTC	Merging Test Criteria
MWIR	Middlewave Infrared
NASA	National Aeronautics and Space Administration
N/A	Not Applicable
NIMA	National Imagery and Mapping Agency
NOAA	National Oceanic and Atmospheric Administration
NPOESS	National Polar-orbiting Operational Environmental Satellite System
N/S	Not Specified
OLS	Operational Linescan System
PD	Property Distance
PDC	Property Distance Clustering
RDR	Raw Data Record
SBRS	Santa Barbara Remote Sensing

SDSM	Solar Diffuser Stability Monitor
SeaWiFS	Sea-viewing Wide Field-of-view Sensor
SNR	Signal-to-Noise Ratio
SRD	Sensor Requirements Document
SWIR	Shortwave Infrared
TBD	To Be Determined
TBR	To Be Reviewed
THEMIS	Thermal Emission Imaging System
TIROS	Television Infrared Observation Satellite
TMI	Type Matching Index
VIIRS	Visible/Infrared Imager/Radiometer Suite
VNIR	Visible/Near Infrared

ABSTRACT

Total cloud cover and layered cloud structure are basic components of an image analysis procedure called "nephanalysis". Many operational users of cloud data require analysis of the extent, type, and physical characteristics of vertically distributed cloud layers. For example, in-flight aircraft refueling has stringent requirements for cloud-free visibility between aircraft at flight altitude. Icing specification and forecasts depend on accurate initial depiction of the constituent particle sizes and state (liquid or frozen) of clouds at specific altitudes. Many other uses, such as accurate prediction of lines-of-sight for aerial reconnaissance, depend on accurate classification of clouds into the classic cloud families. These diverse needs are best met by a novel algorithm called Property Distance Clustering (PDC) that adapts the K-means cluster algorithm by using the local similarity relationships among pixels to combine them in clusters.

This document includes a thorough description of the established behavior of the Cloud Cover/Layers (CCL) Environmental Data Record (EDR) processing approach. This algorithm with initial heritage from K-means clustering is derived primarily from pixel-level intermediate product (IP) inputs such as the Cloud Top Height, the Effective Particle Size, and the Cloud Optical Thickness. The pixel-level cloud mask and cloud phase intermediate products (IP) are used as well. Since both cloud macro (height and phase) and micro (effective particle size and optical thickness) properties are used in characterizing distinct cloud layers, the algorithm can meet the objective of the CC/L EDR to identify the vertical structure of clouds consistent with pixel-level observations within each horizontal (aggregation) cell over a VIIRS image.

The algorithm implements a spatial aggregation method to construct CCL EDR output with approximately constant horizontal cell size. This same aggregation method is also applied to the intermediate products for the other cloud algorithms (cloud optical thickness, cloud effective particle size, cloud top temperature, cloud top height, cloud top pressure, cloud base height) to produce the required outputs for the other cloud EDRs.

1. INTRODUCTION

Techniques for retrieving cloud cover parameters from multispectral satellite imagery have been developed and tested with NOAA/AVHRR and METEOSAT (Arking and Childs, 1985, Desbois et al., 1982). A clustering approach, combined with the threshold method to identify cloud cover, cloud type, cloud top temperature and cloud optical thickness, is shown to be successful. The VIIRS CC/L algorithm is based on the clustering heritage using the derived cloud top height and cloud optical properties to identify cloud layer and cover.

1.1. PURPOSE

This CC/L Algorithm Theoretical Basis Document (ATBD) describes the algorithms used to determine the layered structure of cloud cover using VIIRS EDRs and IPs. These EDRs include Cloud Top Height (CTH), Cloud Optical Thickness (COT) and Cloud Effective Particle Size (EPS). The algorithm also requires the IPs of the VIIRS cloud mask and phase. The primary purpose of this ATBD is to establish guidelines for the production of the CC/L EDR. This document will describe the required inputs, the theoretical foundation of the algorithms, the sources and magnitudes of the errors involved, practical considerations for post-launch implementation, and the assumptions and limitations associated with the products.

1.2. ATBD SCOPE

This ATBD details a novel algorithm to reconstruct, on a pixel basis, the vertical distribution of cloud cover within a Horizontal Cell (HC) from recovered physical parameters. The selected algorithm fits three dimensional cloud cover to a horizontal, vertical and physically consistent cloud structure within each aggregated HC to meet and/or exceed the specification requirements for this EDR. Section 1 describes the purpose and scope of this document; it also includes a listing of VIIRS documents that will be cited in the following sections. Section 2 provides a brief overview of the motivation for the CC/L algorithm, including the objective of the EDR, the currently designed VIIRS instrument characteristics, and the strategy for obtaining the CC/L EDR. Section 3 contains the essence of this document, a complete description of that retrieval process. Consideration is given to the overall structure, the required inputs, a theoretical description of the products, assessment of the error budget, case results from sensitivity studies, practical implementation issues, validation, and the algorithm development schedule. Section 6 provides an overview of the constraints, assumptions and limitations associated with the CC/L EDR, and Section 7 contains a listing of references cited throughout this document.

1.3. VIIRS DOCUMENTS

Reference to VIIRS documents within this ATBD will be indicated by an italicized number in brackets, e.g., *[V-1]*.

[V-1] NPOESS System Specification (SY15-0007).

[V-2] VIIRS Geolocation ATBD (Y3258).

[V-3] NPOESS Calibration and Validation Plan Volume 2: Visible/Infrared Imager/Radiometer Suite (VIIRS) (D31409-02).

[V-4] VIIRS Cloud Mask ATBD (Y2412).

[V-5] VIIRS Cloud Top Parameters ATBD (P1187-TR-I-005).

[V-6] VIIRS Cloud Effective Particle Size & Cloud Optical Thickness ATBD (Y2393).

[V-7] VIIRS Cloud Cover/Layers Performance Test Data Compendium, Supplement to ATBD.

1.4. REVISIONS

PR-08923-04-02, Version 1 Revision 0, Annotated Abstract, June 10, 1998.

PR-08923-04-02, Version 1, Revision 0.1, Annotated Outline, August 15, 1998.

Y2392, Version 1, Revision 2, Cloud Cover/Layers ATBD, October 1998.

Y2392, Version 1, Revision 3, Cloud Cover/Layers ATBD, March 1999.

Y2392, Version 2, Revision 0, Cloud Cover/Layers ATBD, June 1999.

Y2392, Version 3, Revision 0, Cloud Cover/Layers ATBD, May 2000.

Y2392, Version 4, Revision 0, Cloud Cover/Layers ATBD, May 2001.

Y2392, Version 5, Revision 0, Cloud Cover/Layers ATBD, March 2002.

> New version using new PDC/K-Means algorithm.

Y2392, Version 5, Revision 1, Cloud Cover/Layers ATBD, May 2002.

> Incorporates correction for VIIRS sensor scan angle effect on Cloud Cover EDR.

Y2392, Version 5, Revision 2, Cloud Cover/Layers ATBD, January 2004.

> Adds necessary detail for retrieval code implementation. Adds normalization algorithm for cloud optical thickness and effective particle size variables. Removes unnecessary algorithms for cluster mergers and outlier removal.

P1187-TR-I-006, Version 6, Revision 0, Cloud Cover/Layers ATBD, January 2005.

> Describes revisions to k-Means algorithm and implementation of a spatial aggregation method that produces horizontal cells with approximately constant size. Updated to correspond with to NPOESS System Specification rather than VIIRS System Specification.

P1187-TR-I-006, Version 6, Revision 1, Cloud Cover/Layers ATBD, January 2005.

> Adds description of Parallax Correction algorithm. Adds document numbers to references.

P1187-TR-I-006, Version 6, Revision 2, Cloud Cover/Layers ATBD, 16 Jun 2005.

Resolved discrepancies in previous releases (Feb 2005, May 2005). Includes adaption algorithm for new cloud mask that includes designation of multilayer and partial cloudy pixels.

2. EXPERIMENT OVERVIEW

2.1. OBJECTIVES OF CLOUD LAYER RETRIEVALS

The Cloud Cover/Layers EDR requirements are given in the NPOESS System Specification, Appendix D, Section 40.4.2. This algorithm will use IPs that have been retrieved for each image pixel rather than computed from horizontally aggregated EDRs (i.e., EDRs for horizontal cells). The objective is to identify the vertical structure of clouds consistent with pixel-level observations within each horizontal (aggregation) cell over a VIIRS image.

The CC/L product is described in the NPOESS Specification as:

> "Cloud cover is defined as the fraction of a given area on the earth's surface for which a locally normal line segment, extending between two given altitudes, intersects a cloud. The product will also include a binary (cloudy/not cloudy) map indicating the horizontal cells that contain clouds. Day condition for this EDR is when the solar zenith angle is less than 85 deg.
>
> "This EDR will be produced from all nominal NPOESS orbits, but the measurement accuracy for a terminator orbit might be degraded due to VIIRS calibration limitations for a terminator orbit. The terminator orbit is not included in computing the maximum local average revisit time."

The binary (cloud/ not cloudy) map is required at a pixel level and is described in *[V-4]* VIIRS Cloud Mask/Phase ATBD. It is an input to this algorithm but is not described further here.

Table 1 lists the detailed EDR attributes for the CC/L product from the NPOESS System Specification.

The requirements specify two sets of Horizontal Cell Size (HCS) attributes: one for a fine product (HCS is 6 km at nadir) and one for a moderate resolution product (HCS is 25 km worst-case).

The measurement accuracy and precision requirements are set at 0.07 and 0.1 HCS respectively at Nadir and EOS. These requirements are interpreted to apply only to the total cloud cover within a HC, not for separate layer in the same HC. This is because for each layer in the same HC, the layer cloud fraction could physically be as small or smaller than 0.1. There are no cloud cover accuracy or precision requirements for each individual layer.

Table 1. EDR Attributes for Cloud Cover/ Layers

Paragraph	**Subject**	**Specified Value**	**NPP Exclusion**
	a. Horizontal Cell Size (HCS)		
40.4.2-1	1. Moderate, Edge of Swath [VIIRS Guarantee]	25 km	
40.4.2-12	2. Fine, Nadir	6 km	
40.4.2-2	b. Horizontal Reporting Interval [VIIRS Guarantee]	HCS	
40.4.2-3	c. Vertical Reporting Interval [VIIRS Guarantee]	4 Layers	
40.4.2-4	d. Horizontal Coverage [VIIRS Guarantee]	Global	
40.4.2-5	e. Vertical Coverage [VIIRS Guarantee]	0 - 20 km	

Paragraph	Subject	Specified Value	NPP Exclusion
40.4.2-6	f. Measurement Range [VIIRS Guarantee]	0 - 1.0 HCS Area	
	g. Measurement Accuracy		
40.4.2-7a	1. At Nadir, Fine HCS	0.07 HCS Area	
40.4.2-7b	2. At Edge of Swath, Moderate HCS	0.1 HCS Area	
	h. Measurement Precision		
40.4.2-8a	1. At Nadir, Fine HCS	0.07 HCS Area	
40.4.2-8b	2. At Edge of Swath, Moderate HCS	0.15 HCS Area	
40.4.2-9	i. Mapping Uncertainty, 3 Sigma [VIIRS Guarantee]	1.5 km	
40.4.2-10	j. Max Local Average Revisit Time	5.8 hrs	X
40.4.2-13	k. Latency	See Appendix E	
40.4.2-14	l. Binary Map HCS	0.8 km @ Nadir	
40.4.2-15	m. Binary Map Horizontal Reporting Interval	Binary Map HCS	
40.4.2-16	n. Binary Map Measurement Range [VIIRS Guarantee]	Cloudy/Not Cloudy	
	o. Binary Map Probability of Correct Typing		
40.4.2-17a	1. Day, Ocean, OD = 0.5 or Less [VIIRS Guarantee]	92%	
40.4.2-17b	2. Day, Ocean, OD > 0.5 [VIIRS Guarantee]	99%	
40.4.2-17c	3. Day, Land, OD = 1 or Less [VIIRS Guarantee]	85%	
40.4.2-17d	4. Day, Ocean, OD > 1 [VIIRS Guarantee]	93%	
40.4.2-17e	5. Night, Ocean, OD = 0.5 or Less [VIIRS Guarantee]	90%	
40.4.2-17f	6. Night, Ocean, OD > 0.5 [VIIRS Guarantee]	96%	
40.4.2-17g	7. Night, Land, OD = 1 or Less [VIIRS Guarantee]	85%	
40.4.2-17h	8. Night, Land, OD > 1 [VIIRS Guarantee]	90%	
40.4.2-18	p. Degraded Daytime Measurement Condition: Sun Glint < 36 deg [VIIRS Degradation]	Night Performance	
40.4.2-19	q. Excluded Measurement Condition: Aerosol Optical Thickness > 1.0 [VIIRS Exclusion]		

This document also describes the production of aggregated EDR horizontal cells (HCs). These requirements are summarized in Table 2 (CC/L is repeated for convenience). The algorithm described in this ATBD addresses the producing all EDRs on a single HC with HCS ~6 km. For other than the CC/L EDR, each EDR output file includes an average for that parameter (e.g. cloud top temperature) for all cloudy pixels in the HC and separately averaged for each layer identified in the CC/L algorithm. This approach meets or exceeds the EDR specifications as the HCS is better than the moderate resolution requirements and the data provides up to 4 layers and total. This exceeds the specification that requires various averages for total only (all layers grouped together), up to 2 layers, or up to 4 layers.

Table 2. Summary of HCS and Layer Requirements for Cloud EDRs

Req. #	EDR Name	Acronym	# Layers	Fine or Default Resolution (km)		Moderate Resolution (km)	
				Nadir	Worst Case	Nadir	Worst Case
40.4.1	Cloud Base Height	CBH	2 (highest and lowest cloud)	6	10	N/A	N/A
40.4.2	Cloud Cover/ Layers	CC/L	Up to 4	6	N/S	N/S	25
40.4.3	Cloud Effective Particle Size	EPS	Up to 4	5	N/S	N/S	25
40.4.6	Cloud Optical Thickness	COP	Up to 4 [Note A]	5	N/S	N/S	25
40.4.7	Cloud Top Height	CTH	Up to 4	5	N/S	N/S	25
40.4.8	Cloud Top Pressure	CTP	Total only	5	N/S	N/S	12.5
40.4.9	Cloud Top Temperature	CTT	Total only	5	N/S	N/S	25

Note A – NPOESS Specification states that optical thickness is to be derived for each identifiable layer and for the total of all layers.

A derived requirement for this algorithm is to determine a cloud type for each pixel and layer. Cloud type is a required input to the Cloud Base Height (CBH) Algorithm.

The NPOESS System Specification states that the CBH is defined with respect to “sea level where the base occurs.” For CTH, it does not state whether the top height is defined with respect to local sea level or the surface – both have been used in operational systems. We assume that CTH is also defined above sea level for consistency with CBH.

2.2. INSTRUMENT CHARACTERISTICS

The VIIRS instrument will now be briefly described to clarify the context of the descriptions of the CC/L EDR presented in this document. VIIRS can be pictured as a convergence of three existing sensors, two of which have seen extensive operational use at this writing.

The Operational Linescan System (OLS) is the operational visible/infrared scanner for the Department of Defense (DoD). Its unique strengths are controlled growth in spatial resolution through rotation of the Ground Instantaneous Field of View (GIFOV) and the existence of a Low-Level Light Sensor (LLLS) capable of detecting visible radiation at night. OLS has primarily served as a data source for the operational, automated cloud algorithms run by the Air Force Weather Agency (AFWA) and the quality control of these analyses through the manual analysis of the imagery. There have been three automated cloud models used by AFWA: the 3-D Nephanalysis (3DNEPH) Model (Frye, 1976), the Real-Time Nephanalysis (RTNEPH) Model (Hamil et al., 1998) and most recently the SERCCA cloud analyses algorithms (Gustafson et al., 1994) as part of the Cloud Depiction and Forecast System (CDFS) upgrade. These cloud models have been used to produce total and layered cloud cover information for a wide range of operational users across the Department of Defense.

The Advanced Very High Resolution Radiometer (AVHRR) is the operational visible/infrared sensor flown on the National Oceanic and Atmospheric Administration (NOAA) Television Infrared Observation Satellite (TIROS-N) series of satellites (Planet, 1988). Its unique strengths are low operational and production costs and the presence of five spectral channels that can be used in a wide number of combinations to produce operational and research products.

In December 1999, the National Aeronautics and Space Administration (NASA) launched the Earth Observing System (EOS) morning satellite, *Terra*, which includes the Moderate Resolution Imaging Spectroradiometer (MODIS). This sensor possesses an unprecedented array of 36 spectral bands at resolutions ranging from 250 m to 1 km at nadir, allowing a wide range of satellite-based environmental measurements.

VIIRS will reside on a platform of the National Polar-orbiting Operational Environmental Satellite System (NPOESS) series of satellites. It is intended to be the product of a convergence between DoD, NOAA and NASA in the form of a single visible/infrared sensor capable of satisfying the needs of all three communities, as well as the research community beyond. As such, VIIRS will require three key attributes: high spatial resolution with controlled growth off nadir, minimal production and operational cost, and a large number of spectral bands to satisfy the requirements for generating accurate operational and scientific products.

Figure 1 illustrates the design concept for VIIRS, designed and built by Raytheon Santa Barbara Remote Sensing (SBRS). At its heart is a rotating telescope scanning mechanism that minimizes the effects of solar impingement and scattered light. Calibration is performed onboard using a solar diffuser for short wavelengths and a V-groove blackbody source and deep space view for thermal wavelengths. A Solar Diffuser Stability Monitor (SDSM) is also included to track the performance of the solar diffuser. The nominal altitude for NPOESS will be 833 km. The VIIRS scan will extend to 56 degrees on either side of nadir.

The VIIRS SRD places explicit requirements on spatial resolution for the Imagery EDR. Specifically, the Horizontal Spatial Resolution (HSR) of bands used to meet threshold Imagery EDR requirements must be no greater than 400 m at nadir and 800 m at the edge of the scan. This led to the development of a unique scanning approach that optimizes both spatial resolution and Signal to Noise Ratio (SNR) across the scan. The concept is summarized in Figure 2 for the imagery bands; the nested lower resolution radiometric bands follow the same paradigm at exactly twice the size. The VIIRS detectors are rectangular, with the smaller dimension projecting along the scan. At nadir, three detector footprints are aggregated to form a single VIIRS "pixel." Moving along the scan away from nadir, the detector footprints become larger both along track and along scan, due to geometric effects and the curvature of the Earth. The effects are much larger along scan. At around 32 degrees in scan angle, the aggregation scheme is changed from 3x1 to 2x1. A similar switch from 2x1 to 1x1 aggregation occurs at 48 degrees. The VIIRS scan consequently exhibits a pixel growth factor of only 2 both along track and along scan, compared with a growth factor of 6 along scan which would be realized without the use of the aggregation scheme. Figure 3 illustrates the benefits of the aggregation scheme for spatial resolution.

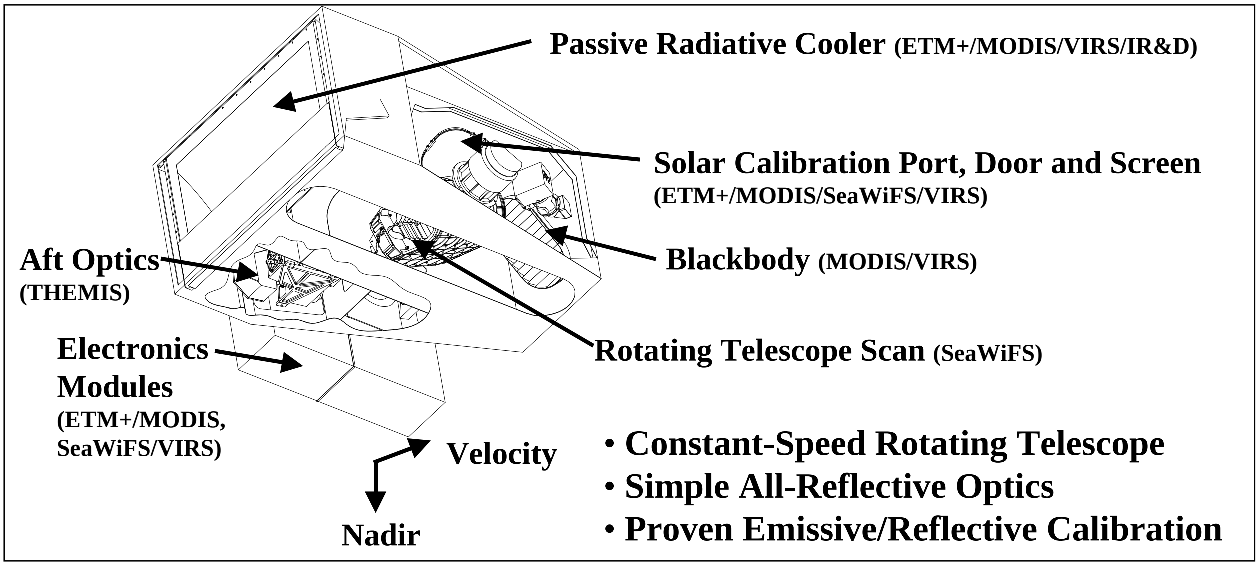

Passive Radiative Cooler (ETM+/MODIS/VIRS/IR&D)
Solar Calibration Port, Door and Screen (ETM+/MODIS/SeaWiFS/VIRS)
Blackbody (MODIS/VIRS)
Aft Optics (THEMIS)
Rotating Telescope Scan (SeaWiFS)
Electronics Modules (ETM+/MODIS, SeaWiFS/VIRS)
Velocity
Nadir
• Constant-Speed Rotating Telescope
• Simple All-Reflective Optics
• Proven Emissive/Reflective Calibration

Figure 1. Summary of VIIRS design concepts and heritage.

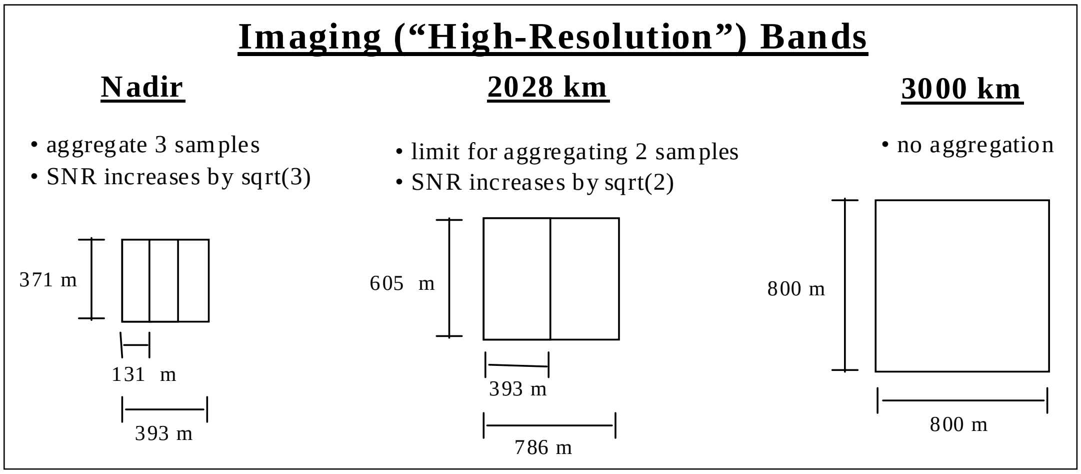

Imaging ("High-Resolution") Bands
Nadir
• aggregate 3 samples
• SNR increases by sqrt(3)
371 m
131 m
393 m
2028 km
• limit for aggregating 2 samples
• SNR increases by sqrt(2)
605 m
393 m
786 m
3000 km
• no aggregation
800 m
800 m

Figure 2. VIIRS detector footprint aggregation scheme for building "pixels." (Dimensions shown are approximate)

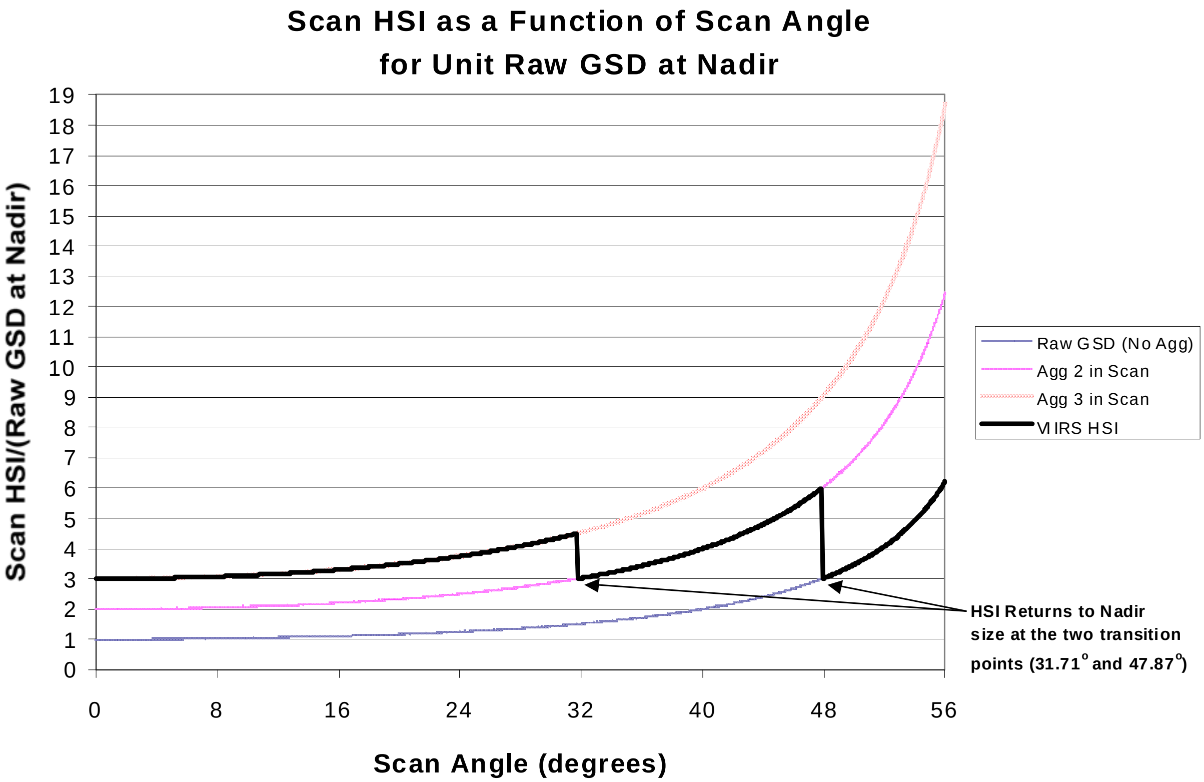


Figure 3. Benefits of VIIRS aggregation scheme in reducing pixel growth at edge of scan.

The VIIRS bands are summarized in Table 3 and Table 4. The positioning of the VIIRS spectral bands is summarized in Figure 4 through Figure 7.

Table 3. VIIRS VNIR bands.

Band Name	Wavelength (μm)	Bandwidth (μm)
Day Night Band	0.700	0.400
M1	0.412	0.020
M2	0.445	0.018
M3	0.488	0.020
M4	0.555	0.020
I1	0.640	0.080
M5	0.672	0.020
M6	0.746	0.015
I2	0.865	0.039
M7	0.865	0.039

Table 4. VIIRS SWIR, MWIR, and LWIR bands.

Band Name	Wavelength (μm)	Bandwidth (μm)
M8	1.240	0.020
M9	1.378	0.015
I3	1.610	0.060
M10	1.610	0.060
M11	2.250	0.050
I4	3.740	0.380
M12	3.700	0.180
M13	4.050	0.155
M14	8.550	0.300
M15	10.763	1.000
I5	11.450	1.900
M16	12.0125	0.950

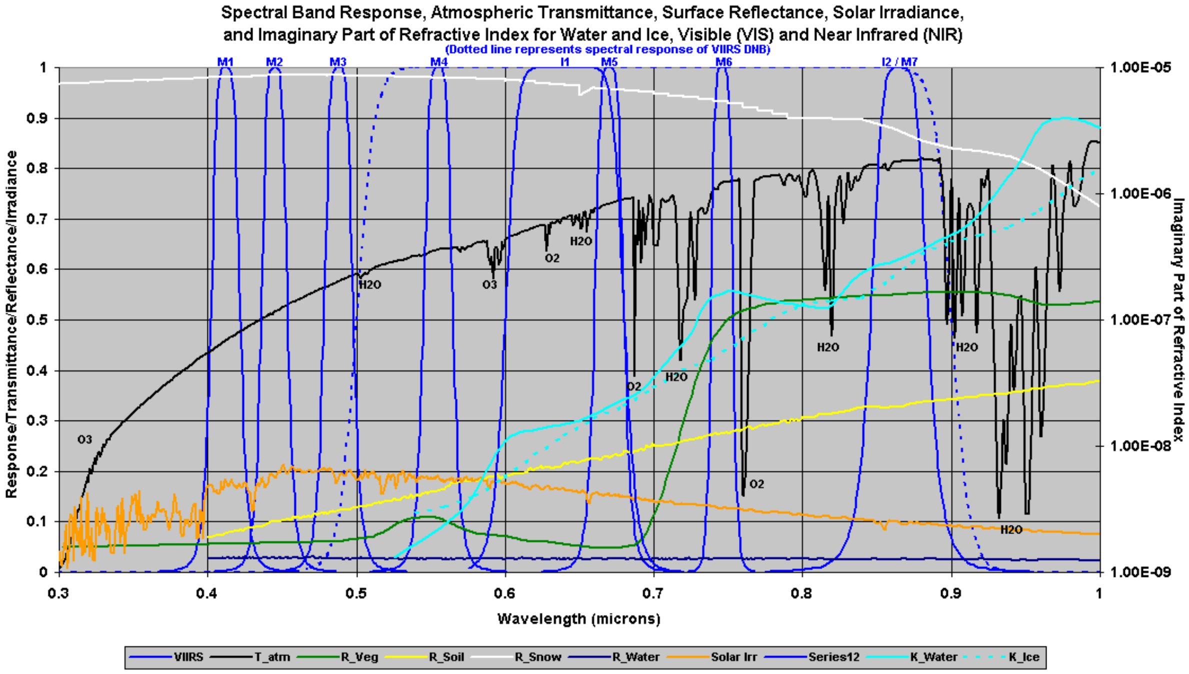


Figure 4. VIIRS spectral bands, visible and near infrared.

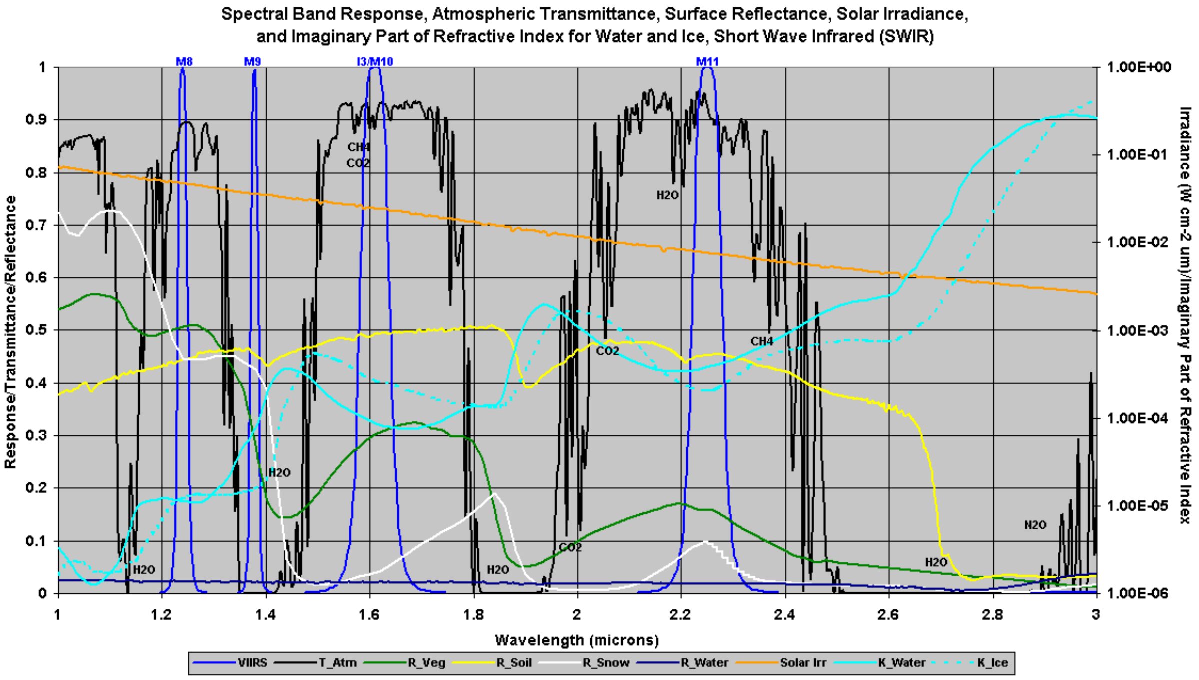


Figure 5. VIIRS spectral bands, short wave infrared.

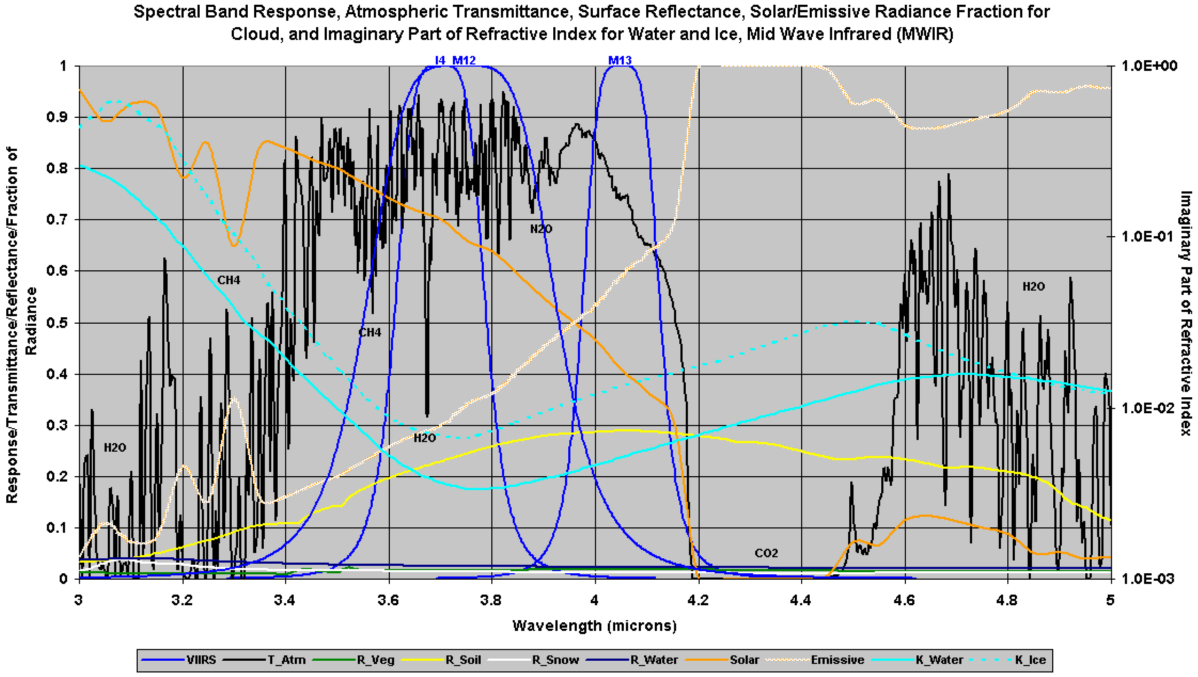


Figure 6. VIIRS spectral bands, medium wave infrared.

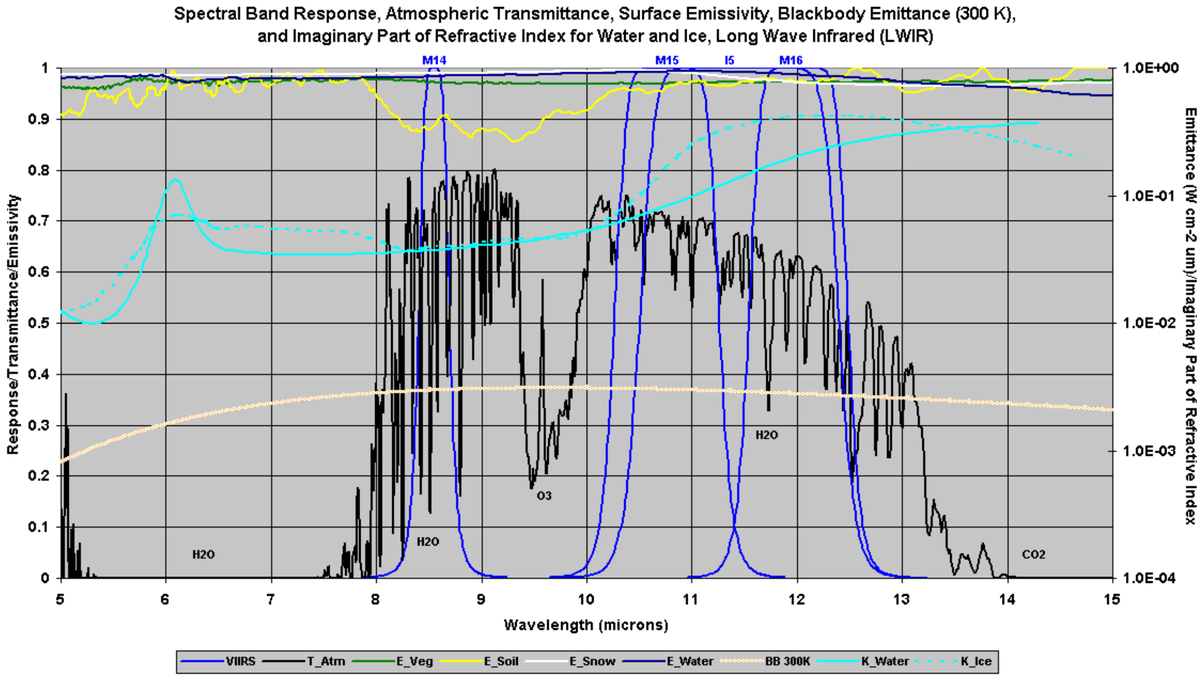


Figure 7. VIIRS spectral bands, long wave infrared.

The VIIRS instrument will exhibit approximately square pixel footprints that increase in size from roughly 750m at nadir to about 1600m at the edge of scan. The lower pixel resolution off-nadir may impair the cloud cover/layers product by offering fewer samples per horizontal cell. The use of a variable Horizontal Cell Size (HCS) with a constant number of pixels per HC might best preserve a consistent CC/L performance from nadir to edge of scan. For example, an 8x8 pixel group produces 6x6 km HCS at nadir and 12.5x12.5 km HCS at edge of scan, which is well within the 25 km requirement. However, this approach does not address the scan geometry of the VIIRS sensor (i.e., bow-tie effect). Instead a strategy that imposes a nearly constant HC of 6 km from nadir to edge of scan is presented in Section 2.5. This approach results in EDRs that are continuous in appearance (i.e., bowtie affect removed) and are not married to any complicated map projection.

Additionally, the off-nadir pixels will have been sensed at oblique look angles. In this case, adjacent cloud elements will tend to obscure cloud-free regions between them, increasing the measured amount of cloud in each horizontal cell. Finally, the edges and sides of larger clouds will be sensed more preferentially, and these may show as spurious small cloud layers in the analysis. This problem is addressed with a statistical correction discussed in Section 4.2.2.4.

2.3. OVERVIEW OF CLOUD PROCESSING MODULE

The overall processing flow of the VIIRS Cloud Module is illustrated in Figure 8, which shows each of the five “Units” comprising the Cloud Module. This document describes the Cloud Cover/ Layers and Grid Cloud EDRs Modules. The overall process is summarized below for context. The diagram indicates the primary outputs of each stage of processing. Pixel level products are at the sensor M-band resolution (~0.75 km at nadir). Gridded products are constructed by aggregating pixels for a horizontal cell size consistent with the EDR product specification.

First, the VIIRS cloud mask is computed. This identifies cloudy and clear pixels and the cloud phase (partly cloudy, water, mixed, cirrus, opaque ice, and overlapping cloud). The cloud mask provides four levels of confidence in the cloud versus clear designation. In increasing order of likelihood a pixel is cloudy the three levels are: confident cloudy, probably cloudy, probably clear, confident clear.

Next, the Cloud Optical Properties (COP) Unit derives the cloud effective particle size and cloud optical thickness. For all cases except daytime water phase clouds it also derives cloud top temperature for each pixel. Then the Cloud Top Properties (CTP) Unit derives the cloud top pressure and height for all clouds. For daytime water phase clouds, it also determines the cloud top temperature.

Thus far, (through the CTP Unit), all the EDRs have been computed on a pixel basis in the satellite projection. This corresponds to ~0.75 km resolution at nadir degrading to about 1.6 km resolution at the edge of scan. The geolocation (latitude and longitude tag) for each pixel is the intersection of the satellite line of sight with the local terrain. The NPOESS Specification requires that the geolocation of the EDRs be based on the latitude/ longitude of the earth below the cloud (determined by the local vertical). The Perform Parallax Correction (PPC) Unit performs this correction.

The Cloud Cover/ Layers (CC/L) Unit groups cloud pixels with horizontal cells into layers. It then outputs fractional coverage for each layer and a total for all layers in each HC. For later use, it also assigns each pixel to a layer. Cloud type is determined for each layer and then for each pixel (based on the pixel layer assignments). This module outputs both a pixel level IP and gridded product IP. The later is at the required horizontal cell size (HCS) for the cloud EDRs.

The Cloud Base Height Unit determines the cloud base height for each cloud pixel. The layer information from the CC/L algorithm is used in this calculation.

Finally, the Grid Cloud EDRs (GCE) Unit inputs the pixel level IPs from all the cloud units and outputs the horizontal cell-averaged values for the cloud EDRs: COT, EPS, CTT, CTP, CTH and CBH. These values are computed as a total averaged over all cloudy pixels in the cell and for each layer. The cloud cover fraction of all layers and individually for each layer is also output.

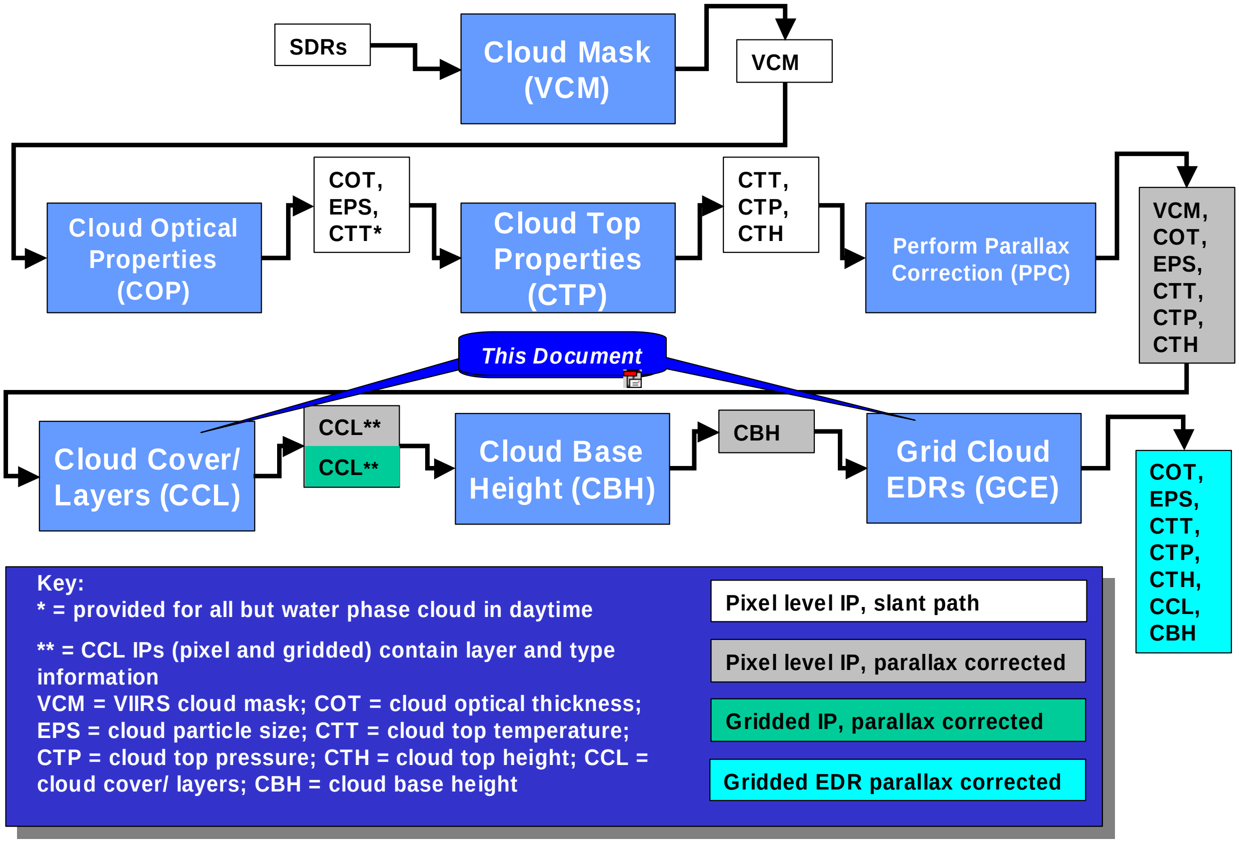

SDRs
Cloud Mask (VCM)
VCM
Cloud Optical Properties (COP)
COT, EPS, CTT*
Cloud Top Properties (CTP)
CTT, CTP, CTH
Perform Parallax Correction (PPC)
VCM, COT, EPS, CTT, CTP, CTH
This Document
Cloud Cover/ Layers (CCL)
CCL**
CCL**
Cloud Base Height (CBH)
CBH
Grid Cloud EDRs (GCE)
COT, EPS, CTT, CTP, CTH, CCL, CBH
Key:
* = provided for all but water phase cloud in daytime
** = CCL IPs (pixel and gridded) contain layer and type information
VCM = VIIRS cloud mask; COT = cloud optical thickness; EPS = cloud particle size; CTT = cloud top temperature; CTP = cloud top pressure; CTH = cloud top height; CCL = cloud cover/ layers; CBH = cloud base height
Pixel level IP, slant path
Pixel level IP, parallax corrected
Gridded IP, parallax corrected
Gridded EDR parallax corrected

Figure 8. Cloud Module Processing Flow

2.4. RETRIEVAL STRATEGY – CLUSTERING ALGORITHM

The central objective of the Cloud Cover/ Layers Algorithm is to group (or cluster) cloudy pixels within a cell into up to four separate layers. Cloud top height is the key property used to determine this grouping. Other properties that can be used include: cloud optical thickness, cloud effective particle size and cloud phase.

We employ the k-Means Algorithm, an established mathematical method for clustering points into groups with similar properties (MacQueen, 1967; Selim and Ismail, 1984; Theiler and Gisler, 1997). Our basic implementation is an unsupervised, iterative method that uses a vector of properties for each point. The grouping of the points is governed by a Euclidean distance metric (over the vector of properties for each point). Points that are determined to be close to each other by the selected metric, tend to be grouped together.

A basic k-Means implementation consists of the following steps:

<u>The k-Means Clustering Algorithm</u>

1. [First guess] Initially assign points to clusters
2. Loop until no more classes are split or grouped
    a. [k-Means Iteration] Loop until a convergence criteria is met
        i. Compute the mean of the property vector for each class (or layer in our case)

  ii. Loop over all pixels and assign each pixel to the class with the closest mean using a distance metric. Keep track of how many pixels are assigned to different classes.

 b. [Group splitting/ joining] Determine if clusters should be split or joined

3. Done

Within the general outline there are many variants. Implementations differ can differ in the following: how the first guess is computed (Step 1); selection of the convergence criteria (Step 2a); variables used to define the pixel state (Step 2a-i); form of the distance metric (Step 2a-ii); and if and how clusters are split or joined (Step 2b).

The k-Means Iteration (the iterative loop of Step 2a) is mathematically guaranteed to converge to a local minimum. Algorithm complexities and differences are driven by several considerations:

- avoiding local minima that are far from the desired global minima
- selecting an appropriate set of state variables and distance metric
- improving efficiency

The k-Means Iteration is the heart of the algorithm and is described in more detail below:

<u>The k-Means Iteration</u>

1. [Initialize]Begin with a set of N points (or pixels), X_i for $i = 1 \ldots N$ that are assigned to K classes (or clusters), C_i for $i = 1 \ldots K$. This mapping is indicated by the function:
   $M_s\{X_i\} \rightarrow C_j$ where the subscript s refers to the k-Means iteration step number
   This assignment comes from the First Guess
   The number of points in each cluster is given by $P\{C_k\}$
   Each point is characterized set of M state variables (or properties):
   $X_i = \left(x_i^1 \;\; x_i^2 \; \cdots x_i^M \right)$
   Specify:
    SMAX → maximum number of k-Means iterations
   GMIN → k-means iterations continue if the minimum fraction of points reassigned at each iteration is greater than this value
   $\alpha_i \;\; i = 1 \ldots M$ scaling factors for each of the input variables
   $S_i \;\; i = 1 \ldots M$ variable selection (=1 if variable is used, =0 if variable not used)
2. [Prepare for k-Means Iteration Loop] Set g = 0, the number of points assigned to a different class
   s = 0, the number of iterations of the k-Means clustering step
3. [k-Means Loop] Loop until s > SMAX OR g/N < GMIN (i.e., until the maximum number of iterations is met or fewer than a threshold number of pixels are reassigned "clusters")
   a. Compute the mean state for each class (the K-Means):

$$A_k = (a_k^1 \, a_k^2 \, \ldots a_k^M) = \left(\sum_{\{X_i\} \in C_k} X_i \right) \Big/ P(C_k) \qquad (1)$$

   b. [k-Means iteration step] Loop over all points $i = 1 \ldots N$,
      i. Compute the "distance" to each of the k-Means

$$D_{i,k} = \left\{ \sum_{m=1}^{M} S_i \left(\frac{x_i^m - a_k^m}{\alpha_m} \right)^2 \right\}^{1/2} \text{ for } k = 1 \ldots K \qquad (2)$$

ii. Set the mapping function for this iteration step s

$$M_s\{X_i\} \to C_{k*} \text{ for k* } \ni \ D_{i,k*} = \min\{D_{i,k} \text{ for } k = 1\ldots K\} \qquad (3)$$

i.e., each point is mapped to the cluster to whose mean it is closest

iii. If the mapping of this point is changed from the previous iteration Set $g = g + 1$

c. Set $s = s + 1$

4. Done

The scaling factors are used to control the relative weighting of the various state variables to the final distance and can be thought of as a unit conversion factor. We also permit variables to be selected or deselected with the variable S_i.

Section 3 and 4 provide details on how the k-Means Algorithm is integrated into the overall processing.

2.5. RETRIEVAL STRATEGY – HORIZONTAL CELLS

The objective of the CC/L and GCE algorithms is to provide all cloud EDRs mapped to the same horizontal cells consistent with the following:

- nearly constant horizontal cell size across a scan
- compatible with scan geometry artifacts of VIIRS (e.g. “bow tie effect”, see following discussion)
- gradual variation in layer assignments from cell to cell
- sufficient number of points used in k-Means clustering to provide reliable statistical performance

The process of grouping pixels in to horizontal cells (HCs) is called aggregation in this discussion.

The VIIRS scan geometry and data content for a single M-band scan (16 along-track detector samples) is illustrated in Figure 9. The geometric distortion of the scan swath at high scan angles is shown (although exaggerated as the figure is not to scale). The figure also shows what is called ‘bow-tie deletion.” This is the elimination of some of the scan-to-scan overlapping pixels (to conserve data-rate) on-board the spacecraft before down-linking to the ground or storage on the spacecraft. The scan-to-scan overlap is illustrated in Figure 10, which shows four VIIRS M-band scans.

The sensor scan parameters used is the analysis and design of aggregation are:

1. VIIRS scan extends from –56.059 to +56.059 degrees
2. Along-track horizontal sample interval is 0.742 km at nadir
3. Cross-track horizontal sample interval is 0.259 km at nadir
4. 16 along-track pixels per scan.
5. Sub-Pixel Aggregation
    a. 3-pixel aggregation at scan angles less than 31.59 degrees
    b. 2-pixel aggregation at scan angles from 31.59 to 44.68 degrees

   c. 1-pixel aggregation at scan angles greater than 44.68 degrees

6. Bowtie Deletion
   a. 0 pixels deleted at scan angles less than 31.59 degrees
   b. 2 pixels deleted at scan angles from 31.59 to 44.68 degrees
   c. 4 pixels deleted at scan angles greater than 44.68 degrees

The actual number of pixels in each aggregation angle zone (both before and after the 3:2:1 aggregation whether on-board or on the ground) is given in Table 5.

All calculations of cell sizes are based on the following orbit parameters, 6371 km Earth radius and 833 km orbit altitude and are considered nominal performance.

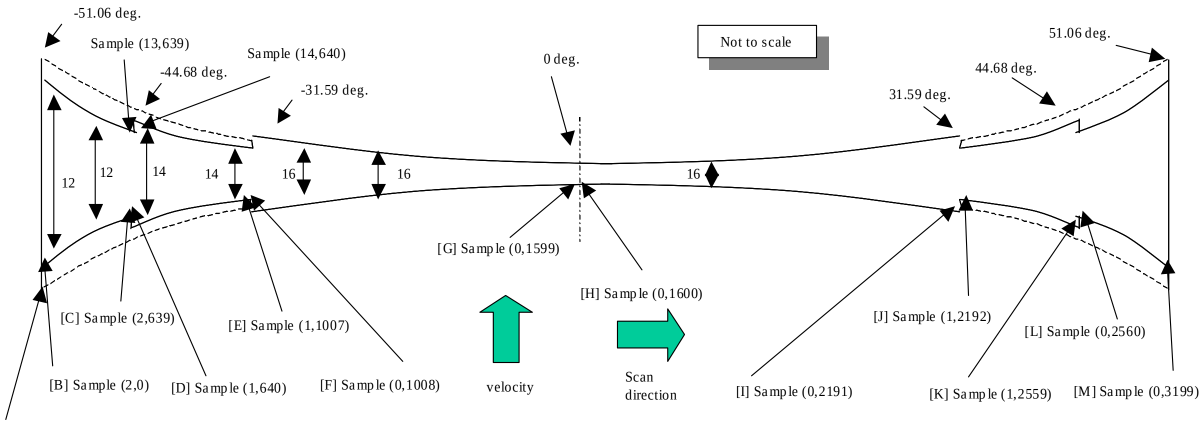


Figure 9. VIIRS Scan Geometry – Single Scan M-Band Channel

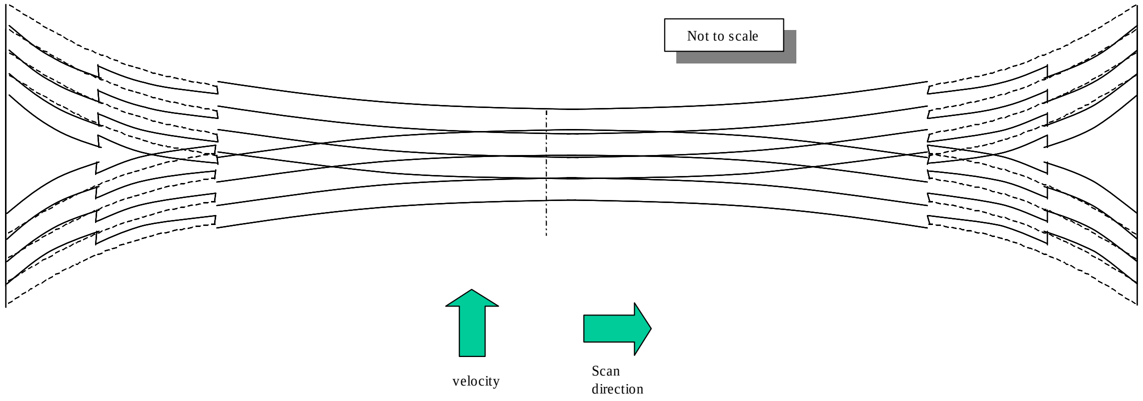


Figure 10. Four VIIRS m-Band Scans

Table 5. Number of pixels in aggregation zones

Zone	unaggregated	aggregated
1. sza < -44.68	640	640
2. -44.68 < sza < -31.59	736	368
3. -31.59 < sza < 0	1776	592
4. 0 < sza < 31.59	1776	592
5. 31.59 < sza < 44.68	736	368
6. sza > 44.68	640	640
Total	6304	3200

The aggregation approach consists of two aspects:

1. Method for formulating the cells for both the CC/L product as well as all other cloud EDRs – we refer to these as "Product Cells"
2. Method formulating the cells over which the statistics are computed for constructing the layers – we refer to these as "Clustering Cells"

The strategies for each of these are discussed in the following sub-sections.

2.5.1. Product Cells

For cloud cover layers, the required horizontal cell size (HCS) is 6 km. The generation of all cloud EDRs on the same reporting interval has many advantages. This would require a small deviation from the current NPOESS Specification for a "fine product" 5 km HCS at nadir for the EPS, COT, CTT, CTH and CTP EDRs. In all cases the performance would be significantly better than the "moderate product" HCS (which ranges from 25 to 12.5 km).

Figure 11 illustrates how 6 km HCS requirement can be met by aggregating (i.e., grouping) numbers of pixels that vary across the sensor scan. The bold solid lines mark each detector. The bowtie deletion limits (at 31.59 and 44.68 degrees) are represented by the vertical dashed lines at which point the detector lines become light. The dashed red lines indicate 6 km distances all across the scan.

The plot shows that some redundancy remains even with bowtie deletion. (Some of this additional redundancy may be eliminated through additional ground-based SDR processing.) Each scan can be divided into two 6 km swaths. At nadir where there is no overlap, 8 pixels are aggregated in the along-track direction to produce the required cell size (i.e., 5.936 km). At end-of-scan, 4 pixels are aggregated and the data in the redundant pixels is ignored. In between, the number of aggregated pixels can be adjusted to give approximately 6 km resolution in the along-track direction. In the cross-track direction, the number of pixels required to produce a 6 km cell will also vary as a function of scan angle. Our HCS calculation also takes into account the varying pixel footprint resulting from the 3:2:1 sub-pixel aggregation.

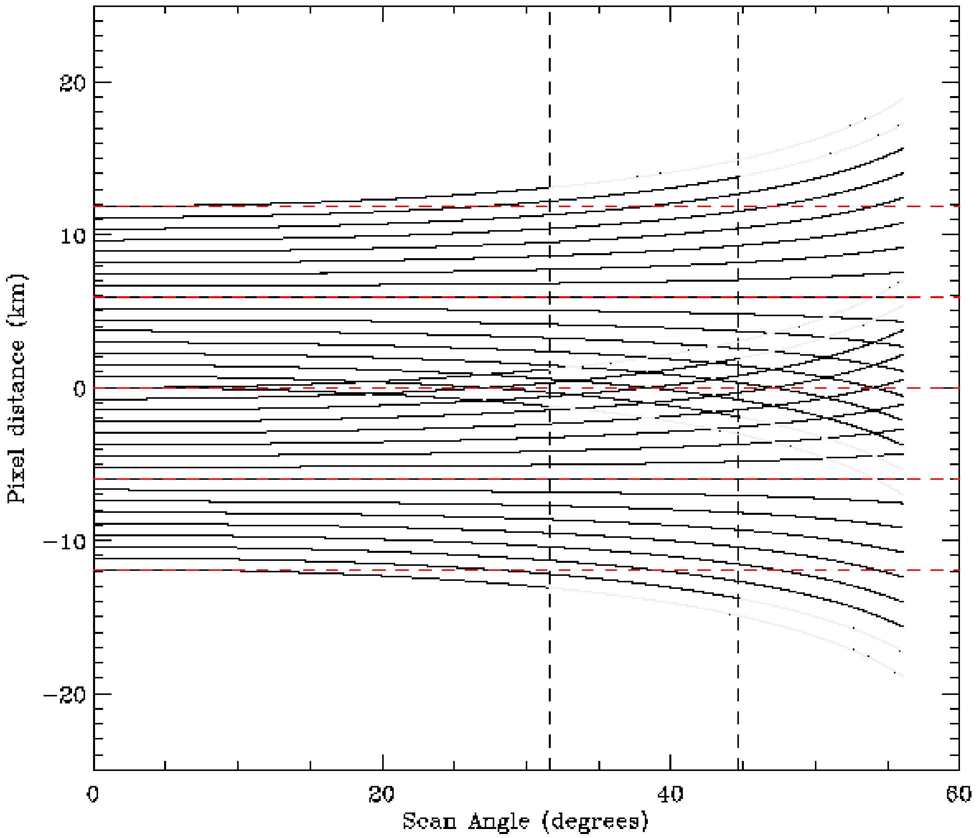


Figure 11. Bow-tie overlap from consecutive scans showing each detector (to scale)

Figure 12 shows the cross-track and along-track cell size, optimized to meet the 6 km requirement, as a function of cross-track distance. There are 254 cells from nadir to end-of-scan. The cells are created through aggregation of pixels in the along-track and cross-track directions. The pixel-aggregation pattern (with cells represented by alternating red and green symbols) is illustrated in Figure 13. The number of pixels in the cell varies across the scan. At nadir, a cell is comprised of 8-by-8 pixels. At end-of-scan, 4-by-4 pixels are used. The discontinuities in the cell size shown in Figure 12 represent transitions in the number of pixels making up a cell and transitions in scale resulting from the cross-track aggregation boundaries. The dimensions of the cells vary between 5.34 to 6.62 km in the cross-track direction and between 5.36 to 6.65 km in the along-track direction, or about ±10%. Each scan is divided into 508 by 2 product cells. At nadir, the HCS is 6.2 km, just slightly in excess of the specification for CC/L fine products (6 km nadir).

In conclusion, the strategy outline above provides identical HCSs for all cloud EDRs of 6 km ±10% with a nadir HCS of 6.2 km. This nearly meets the “fine resolution product” HCS for all cloud EDRs (either 5 km or 6 km at nadir) and vastly improves on the “moderate resolution” product” HCS (that is in the range 25 to 12.5 km for various EDRs). These identical HCSs provides substantial benefits to user who can analyze and display the products on identical grids. It also permits very efficient implementation.

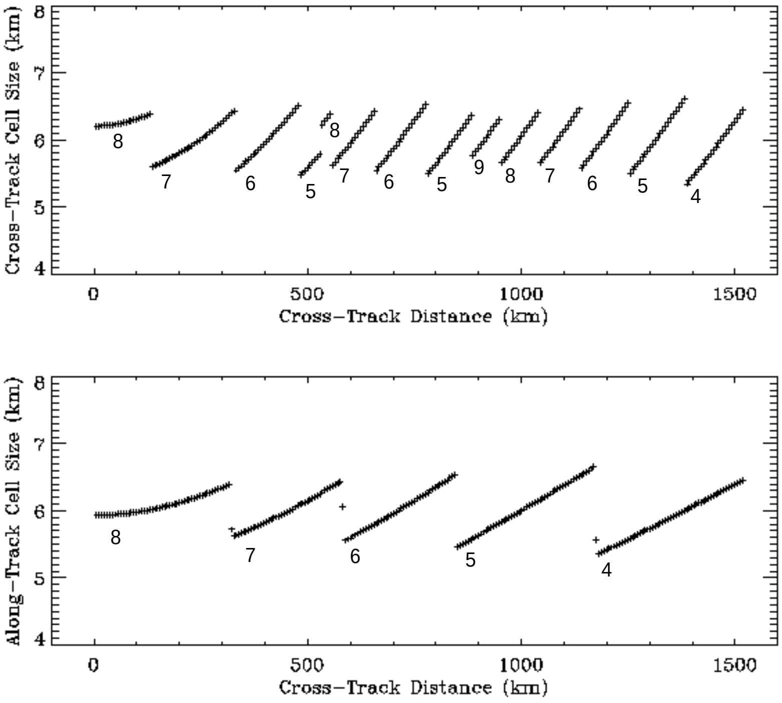


Figure 12. Product cell aggregation optimized to approximately 6 km HCS. Top: Cross-track size; Bottom: Along-track size. Numbers annotating lines indicate number of pixels aggregated in that direction.

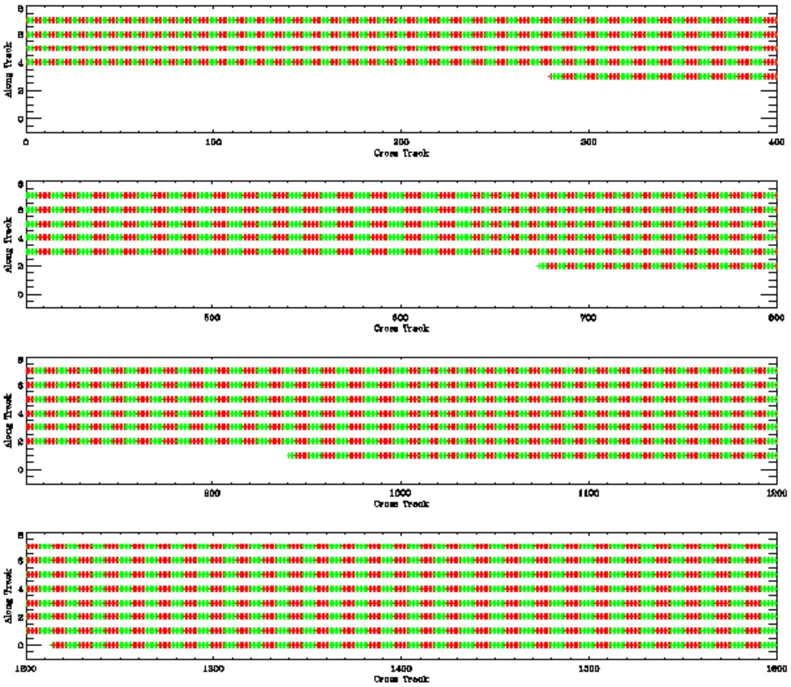


Figure 13. Product cell aggregation (6 km) for ¼ of a scan

2.5.2. Clustering Cells

Our strategy for formulating the cells over which the k-Means clustering statistics are computed (the Clustering Cells) borrows from the current operational Air Force Cloud Depiction and Forecasting System II (CDFS-II). The idea is to derive the statistics and set the layering using cells that are larger than the final product cells and that overlap neighboring product cells. Once the layer grouping over the larger clustering cells are determined, the actual layers are computed by applying the layer partitions only over the small internal cells defining the product HCS.

This approach is illustrated in Figure 14. The top graphic shows a single Clustering Cell and Product Cell pair. The k-Means algorithm is run over all the cloud pixels in the clustering (outer) cell. The products are computed by considering only the actual layer assignments of the product (inner) cell. The bottom graphic shows two adjacent cells and illustrates the overlapping pixels. Note: there is no overlap in the product cell pixels, only those for the clustering cells. A similar overlap occurs in the along-scan direction as well. Consequently, the clusters computed by the k-Means algorithm share some pixels with the clusters computed for each of the adjacent eight cells. On the CDFS-II program, this overlap in clustering cells was determined to be very important to achieve layer assignments that vary gradually from cell to cell.

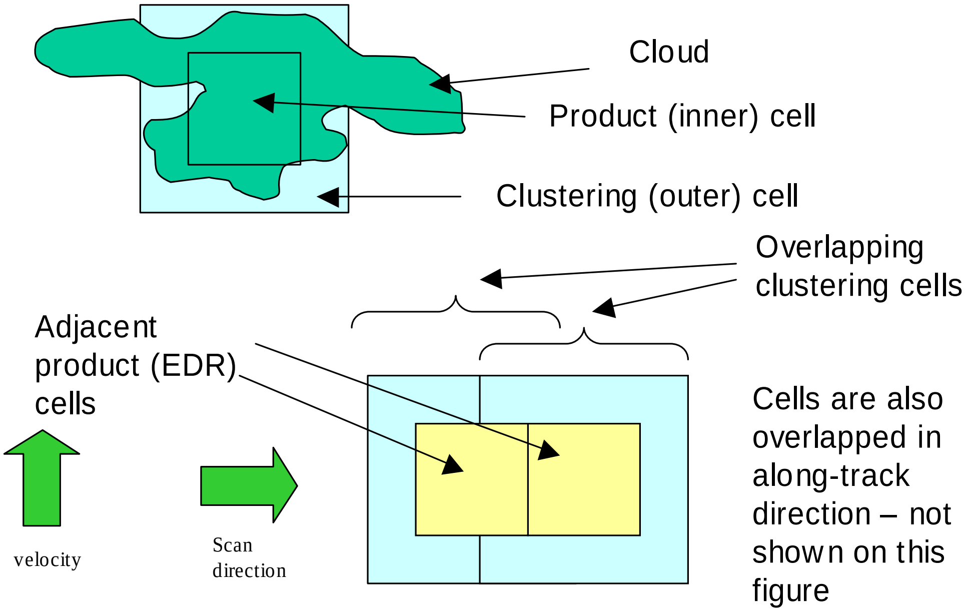

Cloud
Product (inner) cell
Clustering (outer) cell
Overlapping clustering cells
Adjacent product (EDR) cells
Cells are also overlapped in along-track direction – not shown on this figure
velocity
Scan direction

Figure 14. Clustering Cell Approach.

We conducted a trade study (AER, 2005) to determine the optimum size of the clustering cells. This trade resulted in a cluster cell of ~12 x 12 km (twice the size of the product cells). This is based on blocks of 16 x 16 pixels at nadir gradually varying in a manner similar to the product cells to 8 x 8 km at edge of scan (EOS).

Figure 15 compares the number of pixels contained in both the clustering and product cells for CDFS-II (AVHRR and DMSP sensor) and the VIIRS algorithm with a 12 km clustering cell size. As can be seen, the VIIRS algorithm compares favorably to CDFS-II, providing either a greater (most cases) or comparable number of pixels (Edge of scan clustering cell). We thus expect performance no worse than for CSFS-II for this approach to constructing the clustering and product cells. Since CDFS-II applies the layering algorithm after the data have been transformed to a polar stereographic projection, there is a latitude-dependence in the number of pixels. The VIIRS algorithm is independent of latitude (performed in the sensor projection). Also, since only cloudy pixels are employed in the k-Means algorithms these are upper limits on the number of pixels used in the computations.

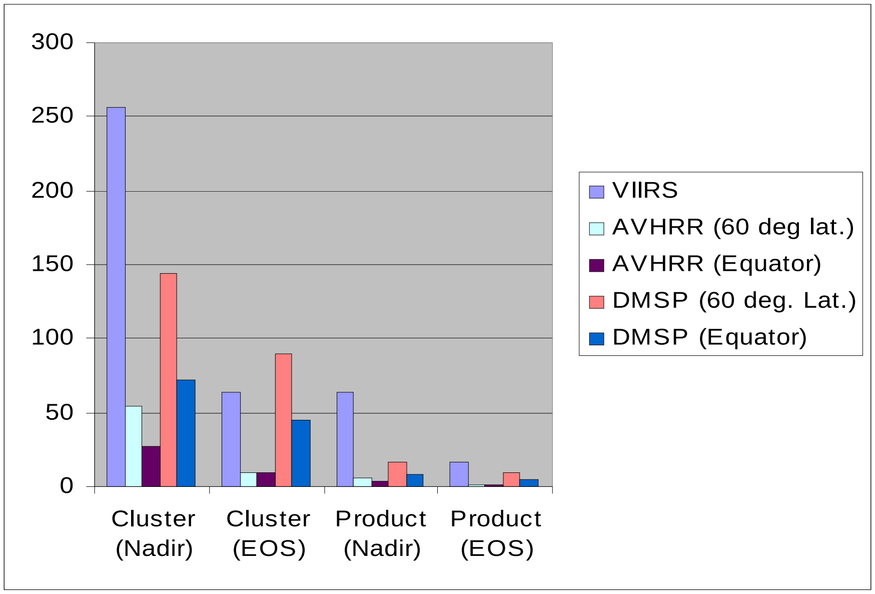


Figure 15. Comparison of Clustering Approach (VIIRS) with CDFS-II Design (DMSP and AVHRR)

2.5.3. CC/L Process Overview

The first step is to group the pixels into horizontal cells of appropriate size that meet the EDR HCS requirements. This grouping accounts for the variation of pixel size with scan and deletion of some but not all pixels that overlap from scan to scan due to the so-called bow-tie effect.

The strategy is to initially partition pixels in a cell into layers based solely on CTH. This is called the first guess. This is followed by application of unsupervised k-Means clustering algorithm (MacQueen, 1967; Selim and Ismail, 1984; Theiler and Gisler, 1997) using a vector of cloud properties for each pixel. This vector is chosen from among the following cloud IPs: CTH, COT, EPS, cloud phase.

The k-Means algorithm iteratively groups and re-groups pixels into sets with similar properties. The sets of pixels with similar properties are determined to form the cloud layers. A weighted Euclidean distance metric is used to determine the pixel groupings (how close each pixel is to a groups mean). In each, pixels may be reassigned from one group (layer) to another. The iterations continue until fewer than a threshold number of pixels are reassigned.

Two variants of this algorithm were implemented. The first method is a variation of the Version 5.3 version of the algorithm and is called the Modified Baseline k-Means (MBKM) Algorithm. The second is a more significant departure designed to address some limitations of the MBKM Algorithm[1]. This second algorithm is called the Extended k-Means (EKM) Algorithm.

The MBKM Algorithm initially assigns each cloudy pixel in a cell to one of four layers based on the pixel's CTH relative to fixed thresholds. The k-Means algorithm is then iteratively applied using a more complete set of cloud properties, chosen from among CTH, COT, EPS, and cloud phase. The iterations continue until less a threshold number of pixels (currently set at 10%) is reassigned (or a maximum number of iterations occurs). This approach fixes the number of layers at four, although it is common for some of the layers to be empty when the processing is complete.

The EKM Algorithm begins by initially assigning all pixels to a single layer. If the CTH standard deviation is greater than a preset threshold, an attempt is made to divide the cell into multiple layers based on the distributions of CTH. Each layer is identified as distinct only if the CTH means and standard deviations meet certain threshold criteria. A maximum of four layers are permitted to be consistent with the NPOESS System Specification. Then the k-Means algorithm is applied using a more complete set of cloud properties (as above chosen from among CTH, COT, EPS and cloud phase). The same stopping criteria are used as with the MBKM Algorithm.

At the end of either of these two variants, each cloudy pixel is assigned to one of up to four layers for each cell. These assignments are used to compute the cloud fractional coverage of each layer as well as fractional coverage of all layers combined.

Each non-empty layer is assigned to one of 5 cloud types: stratus; alto-cumulus, alto-stratus, cumulus, cirrus, or cirrocumulus. This assignment is based on comparing the average properties of each layer with a set of prototypical property vectors for each of the cloud types. A weighted Euclidean distance metric is used. The property vector has the following parameters: CTH, COP, EPS. Cloud phase is also used to determine the subset of possible cloud types appropriate for each layer.

A correction is then applied to account for the difference between line of sight apparent cloud fraction and vertically-projected cloud fraction. The later is the quantity required by the NPOESS System Specification. This effect arises from the finite vertical extent of the clouds and variations in cloud heights within and between layers. The correction is, of necessity, empirical in nature.

Finally the data are output in the form of IPs: both a pixel level IP and a gridded product.

[1] These limitations result from design features common with the Version 5.3 Algorithm

3. ALGORITHM DESCRIPTION

This section describes the two algorithms covered in the ATBD:

- Cloud Cover/ Layers (CC/L) Algorithm
- Grid Cloud EDRs (GCE) Algorithm

The algorithm flow, inputs and outputs are given. Section 4 provides the theoretical and mathematical background of the algorithm steps.

3.1. CC/L ALGORITHM

Figure 16 shows the high-level flow diagram for the CC/L EDR algorithm. The steps are summarized below:

- Assign pixels to HCs – map pixels in the input data to each of the horizontal cells and identify the pixels as belonging to the product cell or cluster cell.
- Apply k-Means algorithm – this includes the first guess at layer assignments (based on the MBKM or EKM approach) and the refinement of the layers based on the k-means state variables.
- Determine cloud type – the mean CTH, COT, EPS, and phase for each layer are compared to empirical values for different cloud types
- Compute layer fractions – total cloud fraction and that for each layer are computed
- Correct for scan angle – cloud fraction is corrected for line-of-sight effects based on a statistical approach

The CC/L algorithm operates on the VIIRS cloud mask and assigns cloudy pixels to various layers by selecting small groups of pixels that share common physical parameters. These parameters include CTH, phase, COT and EPS. These four parameters permit discrimination via different altitudes and types of clouds. The resulting small layers are statistically merged as needed into a layered structure (initially sorted by cloud top height and phase). A pre-defined cloud type model is then used to determine the type of cloud in each layer. Its fractional coverage can be estimated from the population of cloudy (identified by cloud mask) and total pixels within each layer of the HC.

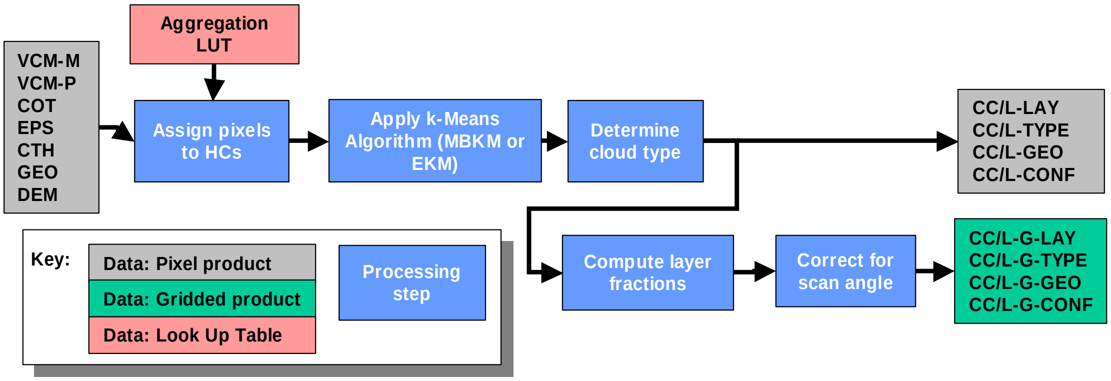


Figure 16. CC/L EDR processing flow

The algorithm inputs and outputs are given in Table 6 and Table 7. Type indicates whether the data are for pixels (M-Band resolution) or gridded (horizontal cells at approximately 6 km HCS).

Table 6. Inputs to CC/L Algorithm

File	Parameter	Acronym	Type	Requried?	Comments
VIIRS SDR	Geolocation (lat/ lon)	GEO	Pixel	x	Only latitudes and longitudes in the SDR file are used and copied to output
VIIRS Cloud Mask (VCM)	Cloud mask	VCM-M	Pixel	x	
	Cloud phase	VCM-P	Pixel	x	
Cloud Optical Properties IP	Cloud Optical Thickness	COP	Pixel		
	Cloud Effective Particle Size	EPS	Pixel		
Cloud Top Properties IP	Cloud Top Height (CTH)	CTH	Pixel	x	

Table 7. Outputs from CC/L Algorithm

File	Parameter	Acronym	Type	Comments
CC/L IP	layer	CC/L-LAY	Pixel	
	cloud type	CC/L-TYPE	Pixel	
	geolocation	CC/L-GEO	Pixel	
	quality level	CC/L-CONF	Pixel	
CC/L Gridded IP	fraction layers for 1 to 4 and total cloud fraction (all layers combined)	CC/L-G-LAY	Gridded	Some layers may be empty (in which case set to fill value)
	cloud type by layer	CC/L-G-TYPE	Gridded	
	geolocation	CC/L-G-GEO	Gridded	For centroid of product cell
	quality level	CC/L-G-CONF	Gridded	

The assignment of input pixels to the product cells and clustering cells is specified in an Aggregation Look-up Table, also shown in Figure 16.

3.2. GRID CLOUD EDRS ALGORITHM

Figure 17 shows the processing flow for the GCE algorithm. Four separate gridded EDR output files are produced containing from one to three EDRs each. The steps are summarized below:

- Geopotential to geometric height conversion
- Compute layers averages for IPs – for CTH, CTT, CTP, COT, EPS, CBH
- Compute average over all layers - for CTH, CTT, CTP, COT, EPS, CBH
- Copy gridded CC/L IPs

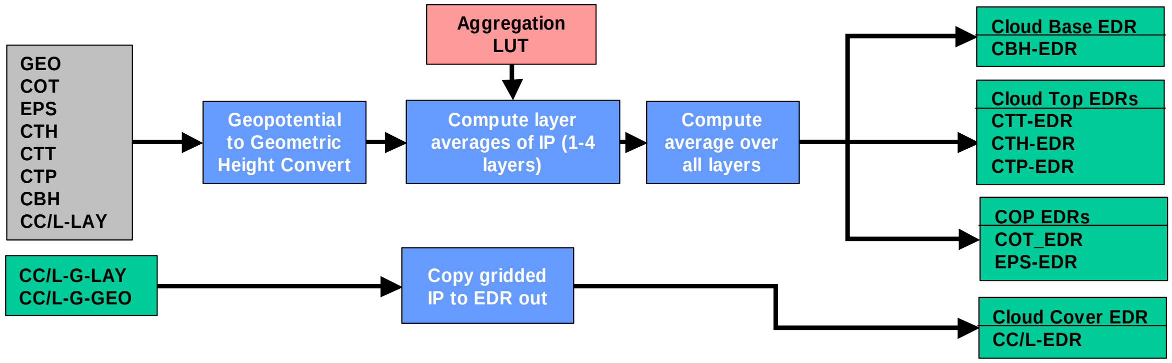


Figure 17. Grid Cloud EDRs processing flow (geolocation and confidence level outputs not shown)

Inputs and outputs are listed in Table 8 and Table 9, respectively. The GCE module uses the same aggregation table specified for CC/L to define the HCs. Geolocation data (i.e., latitude and longitude) are output for each HC.

Table 8. Inputs to GCE Algorithm

File	Parameter	Acronym	Type	Required?	Comments
VIIRS SDR	Geolocation	GEO	Pixel	x	
Cloud Optical Properties IP	Cloud optical thickness IP	COT	Pixel	x	
	Cloud effective particle size IP	EPS	Pixel	x	
Cloud Top Properties IP	Cloud top temperature IP	CTT	Pixel	x	
	Cloud top height IP	CTH	Pixel	x	
	Cloud top pressure IP	CTP	Pixel	x	
Cloud Base IP	Cloud base height IP	CBH	Pixel	x	
Cloud Cover/ Layers IP	CC/L layers	CC/L-LAY	Pixel	x	
Cloud Cover Layers Gridded IP	CC/L gridded layers	CC/L-G-LAY	Gridded	x	
	CC/L gridded geolocation	CC/L-G-GEO	Gridded	x	

Table 9. Outputs from GCE Algorithm

File	Parameter	Acronym	Type	Comments
Cloud Cover Layers EDR	Fraction layers for 1 to 4 and total cloud fraction (all layers combined)	CC/L-EDR	Gridded	Some layers may be empty (in which case set to fill value)
	Geolocation	CC/L-EDR-GEO	Gridded	
	Quality level	CC/L-EDR-CONF	Gridded	
Cloud Base Height EDR	Average cloud base height of each layer and over all layers	CBH-EDR	Gridded	

File	Parameter	Acronym	Type	Comments
	Geolocation	CBH-GEO	Gridded	
	Quality level	CBH-CONF	Gridded	
Cloud Top Parameters EDR	Average cloud top temperature of each layer and over all layers	CTT-EDR		
	Average cloud top pressure of each layer and over all layers	CTP-EDR	Gridded	
	Average cloud top height of each layer and over all layers	CTH-EDR		
	Geolocation	CTPRM-EDR-GEO	Gridded	
	Quality level	CTPRM-EDR-CONF	Gridded	
Cloud Optical Properties EDR	Average cloud optical thickness of each layer and over all layers	COT-EDR	Gridded	
	Average cloud effective particle size of each layer and over all layers	EPS-EDR	Gridded	
	Geolocation	COP-GEO	Gridded	
	Quality level	COP-CONF		

4. THEORETICAL DESCRIPTION OF THE ALGORITHMS

This section outlines the basic principles and mathematical form for both the Cloud Cover/ Layers and Grid Cloud EDRs algorithms. For convenience we also present a review of the theoretical basis of the parallax correction as implemented in the Perform Parallax Correction Unit shown in Figure 8.

4.1. PARALLAX CORRECTION ALGORITHM

Parallax corrections are performed outside of the Cloud Cover/Layers code. It is performed as a stand-alone post-processor after cloud top parameters are retrieved but prior to processing CC/L.

An oblique view correction is performed to correct observed pixel locations for parallax viewing errors. The pixel location correction is dependent on cloud height, view angles, and measurement location (latitude). Shown in Figure 18 are corrections for clouds located at 2, 5, 10, and 20 km at 27ºN. These corrections are to be applied to the retrieval navigation location of the cloudy layer.

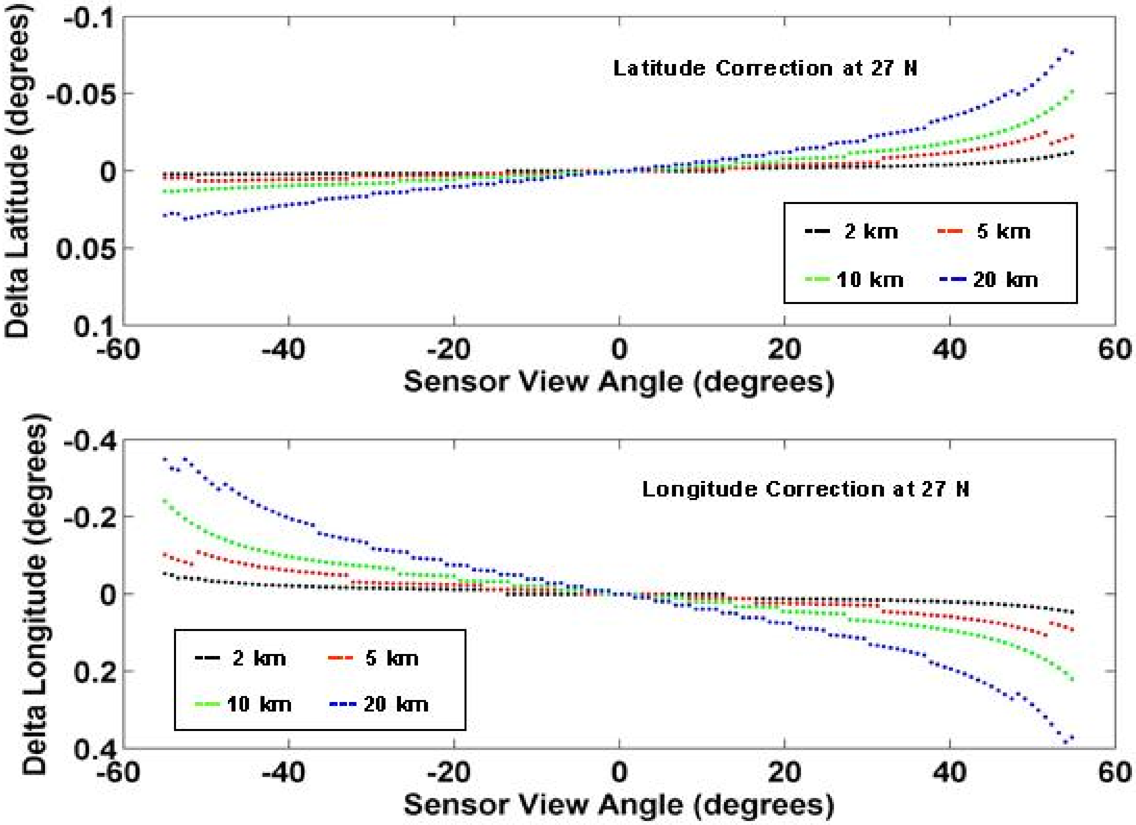


Figure 18. The cloud latitude (upper chart) and longitude (lower chart) correction due to the oblique satellite view when the satellite is located at an average latitude of 27 degrees north. Correction is greatly dependent on cloud altitude.

Given that the spacecraft position vector is available in ECEF coordinates, the pixel position vector is available in geodetic coordinates, and the cloud top height above the ellipsoid is known, only the following eight steps are required by the algorithm:

1. Ingest spacecraft ECEF state vector and EV time tags for current scan line from Geolocation IP.
2. Ingest pixel terrain height corrected geodetic coordinates from SDR.
3. Propagate spacecraft ECEF state vector to pixel time.
4. Transform the pixel position vector from geodetic to ECEF coordinates.
5. Perform vector subtraction to obtain the sensor LOS vector.
6. Solve for the intersection of the sensor LOS vector and the reference ellipsoid plus the cloud top height. This is the cloud position vector in ECEF coordinates.
7. Transform the cloud position vector from ECEF to geodetic coordinates.
8. Find the pixel in the current scanline with the closest geolocation to the cloud.

The following mathematical descriptions of steps four through seven have been derived from the equations in the VIIRS Geolocation ATBD Section 3.3.1.3, ECEF to Geodetic, and Section 3.3.2.2, Basic Earth Ellipsoid Intersection Algorithm.

4.1.1. Propagate spacecraft ECEF state vector to pixel time

The theoretical basis for the algorithm is discussed in the *Theoretical Basis of the SDP Toolkit Geolocation Package for the ECS Project* (ECS Project, 1995), Sections 3.2 and 3.3. The equations are included here for convenience.

The determination of the satellite position vector at a given pixel time is accomplished by the equation:

$$p_i = a_{0,i} + a_{1,i} t_{pixel,norm} + a_{2,i} t^2_{pixel,norm} + a_{3,i} t^3_{pixel,norm},$$

$$t_{pixel,norm} = \frac{t_{pixel} - t_{c,0}}{t_{c,1} - t_{c,0}}$$

where

p_i signifies the position vector component, i, at normalized pixel time $t_{pixel,norm}$,
$a_{k,i}$ are coefficients determined for each vector component,
$t_{c,1}$ is the earth-view scan center time for scan $n+1$,
$t_{c,0}$ is the earth-view scan center time for scan n,
t_{pixel} is the pixel time.

The pixel time, t_{pixel}, is interpolated by the equation:

$$t_{pixel} = t_0 + 2\left(\frac{scanNumber + 0.5}{number\ of\ scans}\right)\left(t_{c,0} - t_0\right)$$

where

t_0 is the earth-view scan start time for scan n,
$t_{c,0}$ is the earth-view scan center time for scan n.

The coefficients are determined by the following set of equations:

$$a_{0,i} = p_i(t_0),$$
$$a_{1,i} = v_i(t_0) \cdot \Delta t,$$
$$a_{2,i} = 3[p_i(t_1) - p_i(t_0)] - \Delta t[2v_i(t_0) - v_i(t_1)],$$
$$a_{3,i} = 2[p_i(t_0) - p_i(t_1)] + \Delta t[v_i(t_0) + v_i(t_1)],$$
$$\Delta t = t_{c,1} - t_{c,0}$$

where

t_0 is the scan start time at scan n,
t_1 is the scan start time at scan $n+1$,
$t_{c,0}$ is the scan center time at scan n,
$t_{c,1}$ is the scan center time at scan $n+1$,
$p_i(t_0)$ is the spacecraft earth-view position vector component at time t_0 ,

$p_i(t_1)$ is the spacecraft earth-view position vector component at time t_1 ,
$v_i(t_0)$ is the spacecraft earth-view velocity vector component at time t_0 ,
$v_i(t_1)$ is the spacecraft earth-view velocity vector component at time t_1 .

Scan times and spacecraft state vector are obtained from the Geolocation IP.

4.1.2. Transform the pixel position vector from geodetic to ECEF coordinates

The relationship between ECEF and geodetic coordinates can be expressed simply in its direct form (NIMA, 1997):

$$x = \left(N + h_{terrain}\right)\cos(lat)\cos(lon)$$ (VIIRS Geolocation ATBD, Equation 3.3-14)

$$y = \left(N + h_{terrain}\right)\cos(lat)\sin(lon)$$ (VIIRS Geolocation ATBD, Equation 3.3-15)

$$z = \left(N\left(1 - e^2\right) + h_{terrain}\right)\sin(lat)$$ (VIIRS Geolocation ATBD, Equation 3.3-16)

$$N = a / \left(1 - e^2 \sin^2(lat)\right)^{\frac{1}{2}}$$ (VIIRS Geolocation ATBD, Equation 3.3-17)

$$e^2 = 1 - \frac{b^2}{a^2}$$ (VIIRS Geolocation ATBD, Equation 3.3-18)

where

(x, y, z) - ECEF coordinates
$(lat, lon, h_{terrain})$ - Geodetic coordinates
N - Ellipsoid radius of curvature in the prime vertical
e - Ellipsoid eccentricity
a - Earth equatorial radius (ellipsoid semi-major axis)
b - Earth polar radius (ellipsoid semi-minor axis)

The geodetic coordinates, *lat*, *lon* and $h_{terrain}$, are inputs from the SDR, and *e*, *a* and *b* are well known physical constants.

4.1.3. Perform vector subtraction to obtain the sensor LOS vector.

$$\vec{u}_{\text{ecef}} = \vec{g}_{\text{ecef}} - \vec{p}_{\text{ecef}}$$

where

$\vec{u}_{\text{ecef}}$ - LOS unit vector in ECEF
$\vec{g}_{\text{ecef}}$ - pixel position vector in ECEF
$\vec{p}_{\text{ecef}}$ - spacecraft position vector in ECEF

The spacecraft position vector is an input from the Moderate Resolution Geolocation IP.

4.1.4. Solve for the intersection of the sensor LOS vector and the reference ellipsoid plus the cloud top height.

The geometry of the problem is shown in Figure 19.

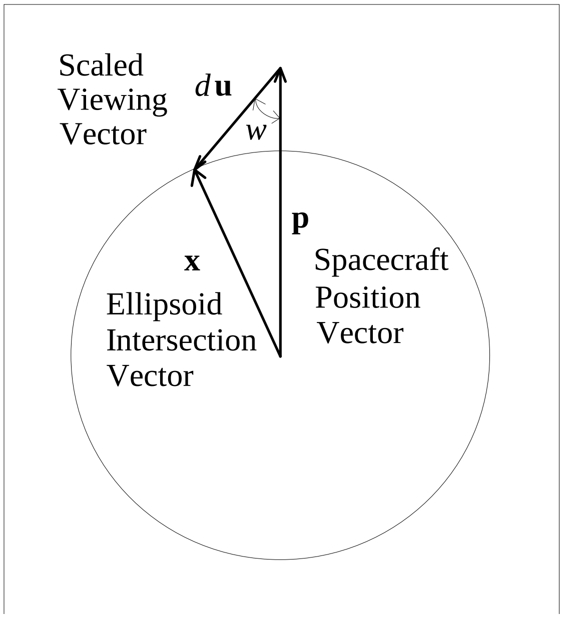


Figure 19. Ellipsoidal Viewing Vector Intersection

Re-scale the viewing vector and satellite vector using the ellipsoid of interest semi-major *a'* and semi-minor *b'* axis dimensions (*a'*, *a'*, *b'*):

$$\mathbf{u}' = \begin{bmatrix} u_1/a' \\ u_2/a' \\ u_3/b' \end{bmatrix} \qquad \mathbf{p}' = \begin{bmatrix} p_1/a' \\ p_2/a' \\ p_3/b' \end{bmatrix}$$ (VIIRS Geolocation ATBD, Equation 3.3-66)

where

$a' = a + h_{cloud}$ and $b' = b + h_{cloud}$

Note that

$$\mathbf{x}' = \begin{bmatrix} x_1/a' \\ x_2/a' \\ x_3/b' \end{bmatrix}$$ (VIIRS Geolocation ATBD, Equation 3.3-67)

where $\mathbf{x}'$ is the unknown cloud position vector (re-scaled).

Solve for the scaling *d* of $\mathbf{u}'$ which intersects the unit sphere:

From the law of cosines,

$$|\mathbf{x}'|^2 = |d\,\mathbf{u}'|^2 + |\mathbf{p}'|^2 - 2\,|d\,\mathbf{u}'|\,|\mathbf{p}'|\cos(w) \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-68)}$$

Using the dot-product, the cosine of the acute angle *w* between $\mathbf{u}'$ and $-\mathbf{p}'$ is:

$$\cos(w) = -(\mathbf{u}'\cdot\mathbf{p}')/(|\mathbf{u}'||\mathbf{p}'|) \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-69)}$$

By definition $|\mathbf{x}'| = 1$, so

$$1 = d^2|\mathbf{u}'|^2 + |\mathbf{p}'|^2 + 2d|\mathbf{u}'||\mathbf{p}'|(\mathbf{u}'\cdot\mathbf{p}')/(|\mathbf{u}'||\mathbf{p}'|) \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-70)}$$

Simplifying and rearranging:

$$d^2|\mathbf{u}'|^2 + 2d(\mathbf{u}'\cdot\mathbf{p}') + |\mathbf{p}'|^2 - 1 = 0 \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-71)}$$

This can be solved for *d* using the quadratic formula:

$$d = \frac{-(\mathbf{u}'\cdot\mathbf{p}') - \sqrt{(\mathbf{u}'\cdot\mathbf{p}')^2 - |\mathbf{u}'|^2(|\mathbf{p}'|^2 - 1)}}{|\mathbf{u}'|^2} \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-72)}$$

This is the smaller of the two solutions for *d*, the intersection closest to the satellite. If the solution is not real, then there is no intersection. This condition will never occur for cloud top heights within spec range.

Use *d* to compute $\mathbf{x}'$ and $\mathbf{x}$:

$$\mathbf{x}' = \mathbf{p}' + d\,\mathbf{u}' \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-73)}$$

$$\mathbf{x} = \begin{bmatrix} x'_1\,a' \\ x'_2\,a' \\ x'_3\,b' \end{bmatrix} = \begin{bmatrix} (p'_1 + d\,u'_1)a' \\ (p'_2 + d\,u'_2)a' \\ (p'_3 + d\,u'_3)b' \end{bmatrix} = \begin{bmatrix} p'_1\,a' + d\,u'_1\,a' \\ p'_2\,a' + d\,u'_2\,a' \\ p'_3\,b' + d\,u'_3\,b' \end{bmatrix} \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-74)}$$

$$\mathbf{x} = \mathbf{p} + d\,\mathbf{u} \qquad \text{(VIIRS Geolocation ATBD, Equation 3.3-75)}$$

4.1.5. Transform the cloud position vector from ECEF to geodetic coordinates.

The theoretical basis for this algorithm is discussed in the *Theoretical Basis of the SDP Toolkit Geolocation Package for the ECS Project* (ECS Project, 1995), Sections 6.4.3 and 6.4.3.3.

4.1.6. Find the pixel in the current scan line with the closest geolocation to the cloud.

1. Determine direction toward scan nadir.

   If pixel_number <= n_pixels/2 increase pixel_number

   If pixel_number > n_pixels/2 decrease pixel_number

2. Initialize minimum distance to a large value.
3. Compute arc length on ellipsoid from current pixel lat/lon to cloud lat/lon using spherical ellipsoid approximation.

   $$range =$$

   $$\arccos(\sin(lat_{pixel})\sin(lat_{cloud}) + \cos(lat_{pixel})\cos(lat_{cloud})\cos(lon_{cloud} - lon_{pixel}))$$

4. Compare arc length to minimum distance.
5. Move one pixel towards nadir.
6. Repeat steps 3 through 5 until minimum distance in found.

4.2. CC/L ALGORITHM

Rather than being a retrieval in the conventional sense of using a physical model to recover unknown parameters from measurements, the CC/L algorithm is statistical in nature. Multiple cloud IPs are used to guide the CC/L PDC algorithm to group high statistically similarity pixels within a single cluster. Unique physical attributes within a cluster/layer are defined since a logical link to classic cloud types is established. CC/L EDR uncertainty is dependent on accurate CTH, COT, EPS, cloud mask and phase IPs.

Figure 16 identifies the five processing steps in the CC/L Algorithm. These are discussed below.

4.2.1. Assign Pixels to Horizontal Cells

The first step is to assign pixels to the clustering and product cells. The design presented in Section 2.5 is used.

The processing model selected is to operate on one primary scan (also called a swath) at a time. Since the clustering cells associated with a particular scan may include pixels found in the previous or following scan, these are also used.

A pre-computed look-up table (LUT) provides a map of all pixels in the current, previous, and next scans into a set of product and clustering cells. This LUT is computed off-line and read in once when the program is loaded. Changes in both types of cells can be easily accommodated provided they meet the following restrictions:

- Product cells cannot cross scans (i.e., they must only use pixels in the “primary” scan and not the previous or next scan)
- Product cells are wholly contained in the associated clustering cell (i.e. no orphan pixels)
- Clustering cells can include pixels outside the current scan but cannot extend beyond the one scan before or after the current scan

The state of each pixel that has been flagged as cloudy in a clustering cell, C, is given as:

$$\left\{X^i\right\} = \left\{x_1^i \; x_2^i \; \ldots x_K^i\right\},\; i \in C$$

For the algorithm implemented here, we employ the following state variables:

$$x_1 \equiv CTH$$

$$x_2 \equiv COT$$

$$x_3 \equiv EPS$$

$$x_4 \equiv CPH$$

CPH is derived from the cloud phase flag in the VIIRS Cloud Mask as follows:

Water phase (includes partly cloudy and water) →	CPH = 0.0
Mixed phase →	CPH = 0.5
Ice phase (includes cirrus, opaque ice, and overlap cloud)→	CPH = 1.0

The collection of state vector variables defined by X above for each clustering cell are used in the k-Means algorithm described in the next section.

The algorithm requires that CTH and CPH be provided for every pixel. In some cases EPS and COT may be missing or invalid. In these cases the values are marked as “fill” and handled as described in the next section.

Pixels identified with the overlapping cloud phase require special consideration in the CCL routine as these pixels represent information from more than one layer. In the current cloud processing chain, the retrieval of COP and CTP treat such pixels as if a single layer of ice cloud was present. As a result the retrieved cloud properties may be unreliable. This problem is addressed in the CCL algorithm by excluding the multi-layer pixels from the process of layer determination. This is accomplished by omitting cluster pixels and re-assigning product plus cluster pixels as product pixels. Additional logic is included to ensure that the number of pixels is sufficient for the layer determination in the case that most of the cell is comprised on multi-layer cloud. If the number of multi-layer cloud pixels exceeds a critical threshold then all pixels are used to determine the layers. This approach ensures that the layers are defined based on pixels representing single layer clouds as long as there is sufficient information to do so.

Special consideration is given to the first and last scans in a data set and the two edges of a scan (at the start or at the end). This is illustrated in Figure 20. Any time we process data where a clustering cell spills over to a missing region, we simply ignore those pixels. As can be seen from the plot at the right of the figure, this leads to only a small reduction in the number of pixels available for the statistical computations and thus should not have a large effect. These edge effects cannot be avoided at the far right and left- edges of a scan. For the first and last scans in a contact, however, this issue can be avoided by a processing strategy that uses the last scans from the previous pass and the first scan from the next contact.

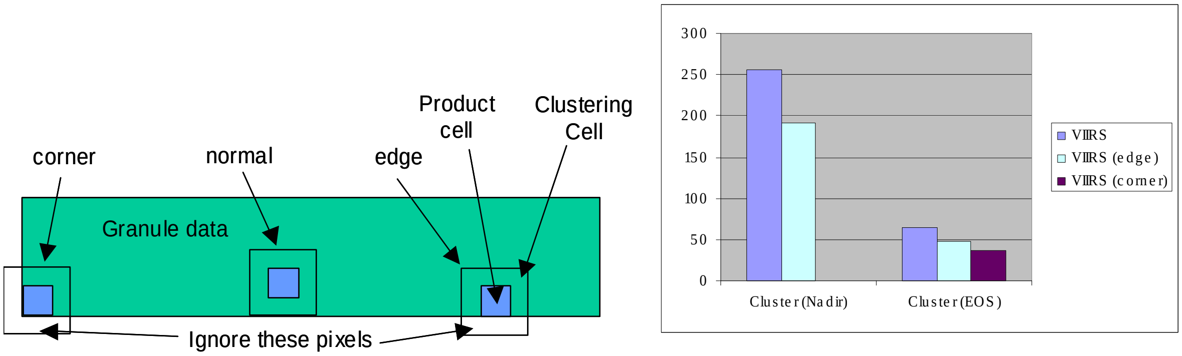


Figure 20. Handling of edges and corners. Left: Illustration. Right: Number of pixels in cluster cells

4.2.2. Apply k-Means Algorithm

Once the pixels for a cluster have been assigned, the next step is to employ the k-Means algorithm to assign each pixel to a layer. We have implemented two alternative algorithm that are summarized below:

- Modified Baseline k-Means (MBKM) algorithm
- Extended k-Means (EKM) Algorithm

The two alternative implementations share a common approach for dealing with the state variables, discussed next.

4.2.2.1 State Variables for k-Means Algorithm

Of the four state variables identified above, each one can be selected or deselected for use in the layering algorithm. In addition, the scaling of each variable (variable α_i of Eq. (3), Section 2.4) can be adjusted in order to allow for greater or lesser influence on the cluster assignments from a given variable.

We experimented with allowing the weights to vary based on the distribution of a variable within a cell. This led to wide variation in influence of the various state variables on the classifications and led to many cases of obviously incorrect layers being assigned. We settled on preset weights (input at run time). These scales represent limits to the variations in cloud properties associated with a single layer and confined within a 6 km cell. Increasing the scaling factor for a given variable results in lesser influence. Decreasing the scaling factor for a given variable results in greater influence. Our current weights are given in Table 10, although these are likely to be refined during algorithm tuning.

Table 10. State Variables and Weights

Variable	**Type**	**Selected?**	**Scaling α_i**
CTH	Required	Yes	2 km
CPH	Required	Yes	0.5
EPS	Optional	Yes	5 µm
COT	Optional	No	N/A

For a variety of reasons, some of these variables may be missing for a given point. This is noted by a so-called “fill value” inserted in the IP. Two variables are mandatory and must be present for a point to be processed: CPH and CTH. If either of these variables is set as a fill value for a given pixel, then that pixel is dropped from the X vector set on which the k-Means Algorithm is applied.

The other two variables (COP, EPS) are optional. If for any of the cloudy points in the cluster set, C, these state variables are missing (and the variable is selected as noted above), then we can ignore the variable in computation of the distance metric for all points in that cluster. (The layer assignments are then based on the remaining variables, e.g., CTH ahd CPH). This approach was taken because COP and EPS can have significant outages, due among other causes to sun glint. These outages will typically cover entire clustering cells, thus preventing any retrieval from being performed. We call this setting: IGNORE_VARIABLE. Alternatively, pixels with missing COP or EPS can be omitted from the X vector set on which the k-Means Algorithm is applied. We call this setting: IGNORE_PIXEL. In regions with significant outages, this option can lead to HCs with inaccurate or missing EDR values.

4.2.2.2 Modified Baseline k-Means Algorithm

The MBKM is a modified form of the Version 5 algorithm. It uses a first guess that is based solely on value of cloud top height relative to fixed thresholds to initially assign pixels to layers (i.e., the first guess). Four partitions, namely Low, Low-mid, High-mid and High Cloud are formed based on CTH boundary values of 2.5 km, 5.0 km and 7.5 km. The pixels are assigned to these first guess partitions independent of their cloud phase. The k-Means iteration is then applied to pixels comprising the Cluster Cell, C. During each iteration of the k-Means Algorithm some number of pixels may be reassigned to different layers. If this number is less than a threshold value (currently 10%), then the algorithm is considered to have converged and the iteration is terminated.

Version 5.3 of the algorithm had further partitioned the initial height ranges into separate sets containing water-, mixed- and ice-phase clouds. There were a total of twelve clusters set by this approach, although some were typically empty. This approach had two problems. First, it often resulted in more than the maximum number of 4 layers being identified. Second, it sometimes led to arbitrary separation of clouds into multiple layers that appear to actually be a mixture of cloud phases (e.g., mixed and ice).

Our current approach still uses phase as one of the state variables. Thus cloud phase still influences the final layering. The current approach does not result in an absolute separation of the cloud phases, however, and layers can contain a mixture pf phases when it makes sense. The relative importance of phase in the classification is controlled by the state vector normalization factors (see Table 10).

Once the first guess is set, the k-Means iteration (Section 2.4) is performed until the convergence criteria are met (less than 10% of the pixels are reassigned or more than 5 iterations).

At this stage, every pixel in the Clustering Cell (outside cell of Figure 14) is assigned to one of four layers. In practice some of these layers may be empty. Subsequent processing is actually only interested in the inner “Product” cell. The following data are saved:

- Pixel based: layer number of each pixel

- Grid based: layer averages of CTH, CPH, EPS and COT (only variables actually used in computation)

4.2.2.3 Extended k-Means Algorithm

Alternately another variant of the k-Means algorithm can be selected, which we call the Extended k-Means (EKM) Algorithm. This variation was developed to address an issue that arose with the MBKM Algorithm: i.e., the algorithm was highly dependent on the first guess. This led to two problems: (1) when the first guess split pixels into two layers with very little difference in CTH, these layers could never be fully merged; (2) when the first guess algorithm identified all pixels within a CTH threshold range as a single layer, these layers could never be divided into two.

The EKM algorithm begins with all cloudy pixels assigned to a single layer. If the standard deviation of CTH is greater than some threshold (e.g., 0.75 km) then an attempt is make to assign the cloudy pixels to two layers. This is accomplished via a Euclidean distance metric with CTH as the only variable. Once the clusters are established the standard deviation of CTH for each distribution is computed, as is the separation between the mean values of CTH. The clusters are accepted as distinct if the separation divided by the sum of the two standard deviations is greater than a prescribed threshold value (e.g., 1.6). Alternatively if this is not the case but the standard deviation of the initial combined group is greater than yet another critical value (e.g., 1.6 km) then the two clusters are also retained. This process is repeated for the cluster with the largest standard deviation in CTH until a maximum of 4 cloud layers is established. The three thresholds used by this algorithm are tunable parameters. The values presented here represent suitable values based on limited testing.

The first guess at cloud layers established based on CTH alone is refined via the k-means approach based on as many as four state variables (CTH, CPH, COT, EPS) in an identical fashion as done with the MBKM method.

4.2.2.4 Scan Angle Effect and Correction for Cloud Fraction

Cloud fractional cover derived from VIIRS measurements increases with increasing scan angle for almost all clouds located at different altitudes. In order to meet the VIIRS cloud fractional cover EDR requirement, this artifact must be removed. J.W. Snow (1986) proposed a single layer cumulus cloud model to correct “apparent cloud cover” – the cloud cover estimated from a particular sensor viewing angle – to local vertical. P. Minnis (1989) used a combination of two cloud fraction data sets derived from nearly simultaneous, collocated Geostationary Operational Environmental Satellite (GOES) West and East radiances, and adapted Snow’s single layer cumulus cloud model to statistically correct the apparent cloud fraction to the cloud fraction that would have been observed at the local vertical. Cloud scenes are not alike, and the viewing angle effect on cloud fraction is highly dependent on cloud size, shape, base-height, thickness, spacing and opacity. Therefore, any successful viewing angle correction scheme will require input of these details. Unfortunately, knowledge of the required cloud parameters is largely unavailable for VIIRS real time EDR processing. An apparent cloud fraction correction scheme must resort to statistical methodology and assumptions that may not be valid for all cloud scenes.

The initial approach to a VIIRS cloud fraction viewing angle correction is to adapt the Snow and Minnis cumulus model, whereby apparent cloud fraction, C, at viewing angle, θ, can be statistically adjusted to local vertical cloud fraction, C_0, through the following relationship:

$$C_0 = C / \{(1+ \sec\theta + \theta\tan\theta)/2\}^{\gamma} \qquad (1)$$

$$\text{Let } F = 1.0 / \{(1+ \sec\theta + \theta\tan\theta)/2\}^{\gamma} \qquad (2)$$

$$\text{Then } C_0 = F\,C \qquad (3)$$

where θ is in radians and γ is defined as the cloud masking exponent used to model complex cloud distributions and properties. It is derived empirically using GOES data in a manner described in detail by Minnis (1989). Following Minnis, γ is dependent on cloud altitude and cloud fraction itself and its empirical values will be described in the next section.

The assumptions that establish equation (1) for the VIIRS baseline cloud fraction angle correction are

1) The clouds are far enough away from the point of observation so that the tangent rays are essentially parallel.

2) Only an off-nadir estimate of cloud fraction is available (i.e., no other multiple viewing angle measurements of cloud cover exist).

3) The clouds are considered totally and uniformly single layer opaque cloud.

4) The VIIRS sensor field of view resolution effect on cloud fraction is assumed to be linear for purposes of resolution correction.

5) The Cumulus model (Snow, 1986) is used as baseline angle correction for cloud fraction.

6) The Masking exponent used in the cumulus model is statistically determined from GOES data sets (Minnis, 1989).

The input requirements for VIIRS cloud fraction angle correction are

1. Initial guess of cloud fraction (apparent cloud fraction is used as the initial cloud fraction estimate),

2. Cloud altitude, h, for low ($h<2$ km); middle (2 km ≤ h ≤ 6 km); and high ($h > 6$ km) cloud classification,

3. VIIRS viewing angle for each processing field of view.

Variables involved in equations 1 to 3 are further explained for clarity as follows:

C: Apparent Cloud Fraction at viewing angle θ.

Note that scan (SA) and viewing angle (VA) hold the following relationship:

$VA = \sin^{-1}\{\sin(SA)*(SH+ER)/ER\}$, where SH is satellite altitude and ER is earth radius.

C_0: Cloud Fraction projected on the Earth's surface at local vertical.

γ: Cloud-masking exponent (in general $0 \leq \gamma \leq 1$; γ=1 implies no cloud-masking; for very small cloud fraction (<0.05), γ can be greater than 1 and may approach 2)

F: cloud fraction viewing angle correction factor -- a function of viewing angle and cloud altitude, due to the dependence of γ on vertical cloud location and cloud fraction itself.

For the cumulus cloud model, cloud-masking exponents (γ) are statistically derived and shown in Table 11.

Table 11. Cloud masking exponents as a function of cloud cover (0.0 to 1.0) and cloud altitude (low, middle, and high clouds).

Cumulus Cloud Model			
γ - Cloud Masking Exponents*			
Cloud Fraction	**Cloud Altitude, H (km)**		
	Low **H < 2**	**Middle** **2 ≤ H ≤ 6**	**High** **H > 6**
0-0.05	**2.019**	**1.402**	**1.446**
0.05-0.1	**1.014**	**0.581**	**0.756**
0.1-0.15	**0.612**	**0.279**	**0.535**
0.15-0.2	**0.508**	**0.167**	**0.468**
0.2-0.4	**0.229**	**0.140**	**0.413**
0.4-0.6	**0.217**	**0.160**	**0.236**
0.6-0.8	**0.139**	**0.067**	**0.138**
0.8-1.0	**0.011**	**N/A**	**0.013**
*After P. Minnis, 1989 (JGR)			

Figure 21 shows the cloud masking exponents for low, middle, and high clouds plotted as a function of cloud fraction (cover). Note that the cloud masking exponent is much larger than 1.0 for very small cloud fraction (cover < 0.05). The curves are the best non-linear fit of the point values.

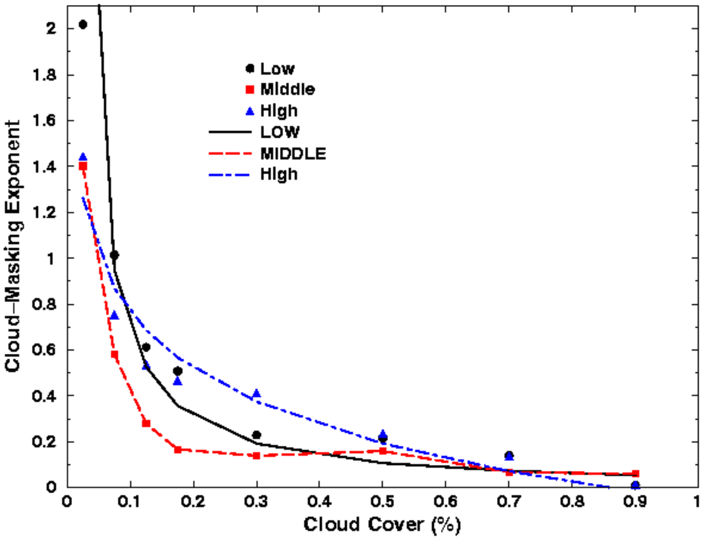


Figure 21. Plots of cloud masking exponents (point values and curves) for low, middle, and high clouds.

Figure 22 are viewing angle correction factors derived from Equation 2 using cloud masking exponents defined in Table 11. For each cloud altitude (low, middle, or high) and initial cloud cover, the correction factor F can be uniquely derived and then used in Equation 3 to adjust apparent cloud cover, C, to cloud cover of local vertical, C_0.

Figure 23 are comparisons of apparent and local vertical cloud cover. Apparent cloud covers are assumed to be constant for all simulated viewing angles (0 to 70 deg). Apparent cloud covers are also varied from low (0.025) to high (0.9) to model the angle correction dependency on cloud cover itself. Obviously, for all clouds, the cloud cover viewing angle effect reaches its maximum at the Edge of Scan (EOS): the larger the viewing angle, the larger the cloud cover correction will be. The amount of correction as a function of cloud fraction and altitude is less intuitive.

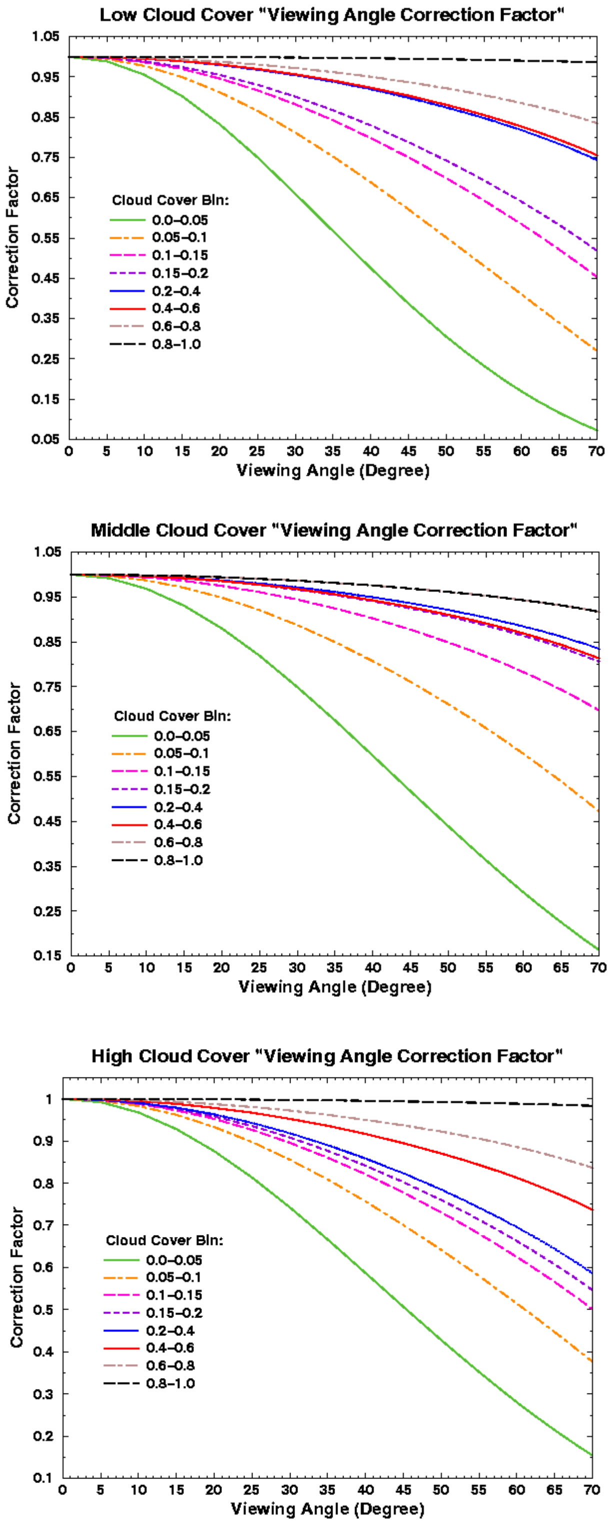
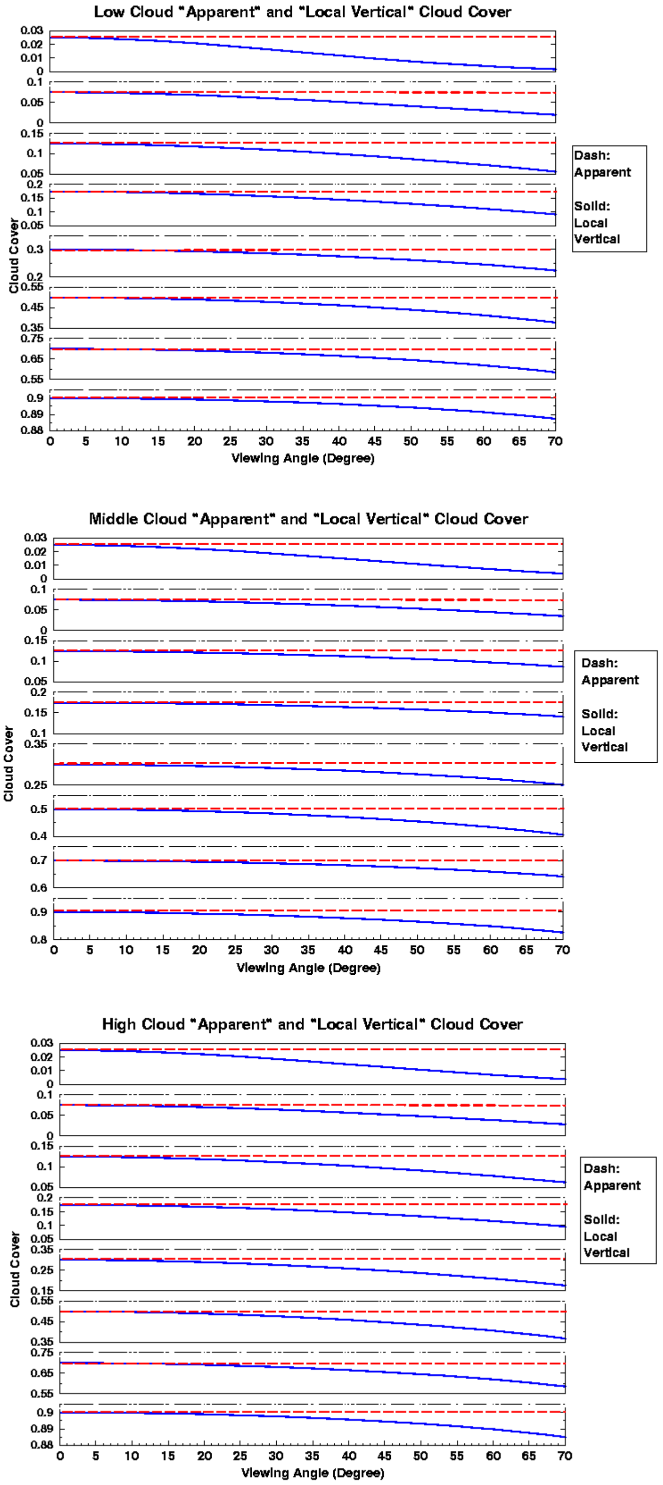


Figure 22. Viewing angle correction factor for low (H < 2 km) (top panel), middle (2 km ≤ H ≤ 6 km) (middle panel) and high (H > 6 km) (bottom panel) cloud cases.



Low Cloud "Apparent" and "Local Vertical" Cloud Cover

Middle Cloud "Apparent" and "Local Vertical" Cloud Cover

High Cloud "Apparent" and "Local Vertical" Cloud Cover

Figure 23. Comparisons of apparent constant cloud covers (0.025; 0.075; 0.125; 0.175; 0.3; 0.5; 0.7; and 0.9) to cloud covers corrected to local vertical for low (H < 2 km) (left panel), middle (2 km ≤ H ≤ 6 km) (right panel), and high cloud (H > 6 km) (bottom panel) cloud cases.

4.2.3. Cloud Type Assignments

Cloud type determination is performed by matching the average value of CTH, COT, and EPS of each cluster/layer with the empirical clouds properties. Hence each distinct layer within each HC will be identified as a cloud type with attributes described in Table 12 (Weickmann and Aufm Kampe, 1953; Heymsfield and Platt, 1984; Dowling and Radke, 1990; Liou, 1992). Each cloud type attribute lists a range of typical values and the specific value used during processing for cloud type determination. For each cloud layer, the range of possible cloud types is determined by the most prevalent cloud phase for a given HC (as layers with mixed phase are permitted by the CC/L algorithm). Once the cloud type for each layer in the HC is established, a pixel-level cloud typing is assigned and included as one of the IPs generated by the CC/L module. The pixel-level cloud typing is required as input in the CBH algorithm.

Table 12. Predefined cloud types characterized in terms of their macro (height and phase) and micro (size and optical thickness) properties. Attribute range in parantheses. Processing value without parantheses.

Cloud Type	Height (km)	Sizes (μm)	Optical Thickness	Phase
Stratus (St, Sc)	(<2.5) 1.3	(2-25) 13.5	(1-10) 5.5	Water
Alto Cumulus/Stratus (Ac, As)	(1.5-5.5) 3.5	(4-30) 17	(2-32) 17	Water/Ice
Cumulus (Cu, Cb)	(0.2-6.5) 3.3	(5-50) 27.5	(3-50) 26.5	Water/Ice
Cirrus (Ci)	(6-12) 9	(10-100) 55	(0.01-5) 2.5	Ice
Cirrocumulus (Cc)	(6-15) 10.5	(30-120) 75	(1-8) 4.5	Ice

4.3. GRID CLOUD EDRS ALGORITHM

4.3.1. Conversion from Geopotential to Geometric Height

The input numerical weather prediction height fields used to determine the cloud top and cloud base heights are given in terms of geopotential height. Thus, the computed CTH and CTP IP values are also in geopotential height. To comply with the EDR specifications for these values (which specify geometric height above mean sea level), we convert the CTH values to geometric height.

We use the following relationship taken from Mahoney[2] (2001):

[2] Mohoney identifies and corrects some inconsistencies and errors in geopotential to geometric height conversion formulas given in the US Standard Atmosphere and the Federal Meteorological Handbook.

$$Z(H,\theta) = \left[1 + \beta_1 \cos(2\theta)\right] H + \left[1 + \beta_2 \cos(2\theta)\right] \frac{H^2}{h*}$$

Z is geometric height above the WGS-84 ellipsoid

H is geopotential height

θ is the geodetic latitude

$\beta_1 = 0.002644$

$\beta_2 = 0.0089$

$h* = 6245$ km

The final EDR is thus given in terms of geometric height above the WGS-84 ellipsoid, which is assumed to represent local mean sea level.

4.3.2. Computation of EDRs

The mathematical form of the computation of the output EDRs is given in this section.

For each of the following EDRs, the average value of the parameter given in the IP is computed individually for each of up to four layers. An average over all pixels (i.e., all layers) is also given.

Let y_i^p = a pixel level (for pixel i) Intermediate Product where p is one of the following products: CTH, CTP, CTT, EPS, COP, CBH (with the geopotential to geometric height correction applied to CTH and CBH). The output EDRs for a given Product Cell is given by:

Let $\left\{C_k^{\mathrm{p}}\right\}$ be the set of valid pixels in the layer k of the product cell for product p

and $\left\{C_*^{\mathrm{p}}\right\}$ be the set of all valid pixels in the product cell for product p

$\mathrm{P}\left\{C_k^{\mathrm{p}}\right\}$ and $\mathrm{P}\left\{C_*^{\mathrm{p}}\right\}$ are the number of points in layer k and all layers respectively

Only those pixels in a product cell that have valid data (i.e. are not fill) are used

$\mathrm{Y}_{\mathrm{k}}^{\mathrm{p}} = \sum_{y_i \in \left\{C_k^{\mathrm{p}}\right\}} y_i^p \Big/ \mathrm{P}\left\{C_k^{\mathrm{p}}\right\}$ is the layer-averaged EDR for layer k

$\mathrm{Y}_*^{\mathrm{p}} = \sum_{y_i \in \left\{C_k^{\mathrm{p}}\right\}} y_i^p \Big/ \mathrm{P}\left\{C_*^{\mathrm{p}}\right\}$ is EDR for the entire cell (i.e. all layers)

The Cloud Cover Layer product are computed in the CC/L Unit and merely copied to the final EDR output files.

4.4. VARIANCE AND UNCERTAINTY ESTIMATES

Errors in the CC/L EDR arise from several sources and in several places. Individual algorithms are sensitive to SDR measurement noise, band-to-band registration errors, input EDR and IP errors, and other effects. Table 13 summarizes possible error sources.

Table 13. Possible error sources for each algorithm.

Algorithm	Affected by SDR Noise?	Affected by Registration Errors?	Affected by IP Errors?	Other Error Sources
Initial CTH/Phase Clustering	Yes	Yes	Yes	Cloud Range Bin
PDC/K Means Clustering	Minimal	Minimal or No Effect	Yes	Centroid/Mean Definition Number of Iteration PDC Definition
Cloud Type Determination	Minimal or No Effect	Minimal or No Effect	Minimal or No Effect	Cloud Type Definition
Cloud Fraction Determination	Yes	Yes	Yes	Cloud mask Definition

4.4.1. Error Budget

For a complete description of the errors that impact the CC/L Algorithm see the Raytheon VIIRS Error Budget, Version 5 (Y3249). Those error budgets are predicated on the linearity and independence of errors. However, in the Cloud Cover/Layers Algorithm, the contributing components are strongly coupled (i.e., non-independent) and act nonlinearly. While these error budgets are based on pixel numbers that are much more numerous that comprising the VIIRS HCs (i.e., 8-by-8 at nadir and 4-by-4 at edge of scan), they are useful for algorithm development and validation.

5. ALGORITHM TESTING AND ERROR ANALYSIS

This section illustrates results from both the CC/L and GCE modules. Two options are provided with the cloud cover layers algorithm. The original algorithm determines the first guess at layer identification based on comparison of CTH to fixed heights. This algorithm then uses a k-means type approach to refine the assignment of the cloud layers. The k-means analysis has been modified relative to the original baseline algorithm to include CTH, EPS, COT, and cloud phase as parameters in the analysis. The algorithm is thus referred to as the Modified Baseline K-Means algorithm (MBKM). The second option included in the CC/L module replaces the first guess based on fixed heights with a first guess routine that is based on a statistical analysis of the CTH. Refinement of the layers is performed using the same k-means algorithm described above. This algorithm is referred to as the Extended K-Means (EKM) algorithm. Results from both the MBKM algorithm and the EKM algorithm are presented in this section.

5.1. DESCRIPTION OF DATA SETS AND SIMULATION

Table 14 summarizes the datasets used in algorithm testing.

The s10_Day_Mid_Lat_Spring scene was used in early stages of algorithm development and to confirm unit tests for the previous version of the CC/L algorithm. A large portion of this scene contains outages where cloud IPs are not produced due to sun glint restrictions.

The AER scenes consist of input IPs derived from MODIS data using in-house equivalents of the VIIRS cloud algorithms. (Note the MODIS products cannot be used directly because all products are not provided at the pixel-level.) Of course the MODIS data differs from VIIRS in its spatial sampling and resolution. We have simulated the 16 lines per scan of the VIIRS sensor by pixel replication of the MODIS 10 lines per scan. However, no correction has been applied to the data in the cross track direction to fully represent the VIIRS scan geometry. Therefore specialized aggregation tables have been developed to process the AER scenes through the CC/L and GCE modules. In this case, tables are supplied corresponding to cell sizes of approximately 5 by 5 km. (Also available are aggregation tables corresponding to cell sizes of 10 by 10 km.)

NGST has provided VIIRS cloud IP data based on radiative transfer (RT) simulations that provide global representation for the cloud products. The analyses of the results from these simulations were not complete at the time this document was prepared. The results will be added however when cloud chain testing is completed by end of 2005.

Table 14. Test Data Sets

Name	Created by	Description
s10_Day_Mid_Lat_Spring	Raytheon	Cloud IPs based on MODIS data with 10 pixels per scan and produced by VIIRS algorithms.
MYD2003183.0750	AER	Cloud IPs based on MODIS data with pixel replication to give 16 pixels-per-scan.
MYD2004296.0410	AER	(see above)
MYD2004296.0810	AER	(see above)
MYD2004296.1910	AER	(see above)
v45A_01_02m_N6	NGST	Cloud IPs derived from simulated VIIRS data with global representation.

5.2. RESULTS FROM THE MYD2003183.0750 TEST SCENE

This section illustrates results from the AER-prepared MYD2003183.0750 test scene. The observations were obtained July 2, 2003 by the Aqua satellite. The location is over North America. The time of day is nighttime. Figure 24 show the MWIR channel data for this MODIS granule.

Hudson and James Bays are visible at the top. The MYD2003183.0750 test scene is derived from a portion of this scene taken from the middle of the granule. All plots presented in subsequent sections are inverted relative to this image. A complete set of the test data is found in the Cloud Cover/Layers Performance Test Data Compendium [V-7].

Results from both the CC/L and GCE retrievals are presented here. For comparison, the CC/L retrievals have been performed using both the MBKM and EKM algorithms. Section 5.2.1 illustrates results obtained with the MBKM algorithm. Section 5.2.2 illustrates results obtained with the EKM algorithm. Cloud cover, cloud type, and cloud top height are presented as examples of the gridded EDRs. Other EDRs (i.e., CTT, CTP, COT, and EPS) are not included for reasons of brevity. Section 5.2.3 demonstrates the performance of the CC/L MBKM algorithm with an example diagnostic plot generated for a selected product cell. Section 5.2.4 demonstrates the performance of the CC/L EKM algorithm with an example diagnostic for the same product cell. Finally Section 5.2.5 summarizes the subjective scoring of the MBKM and EKM algorithm performance based on inspection of diagnostic plots for 100 randomly selected product cells.

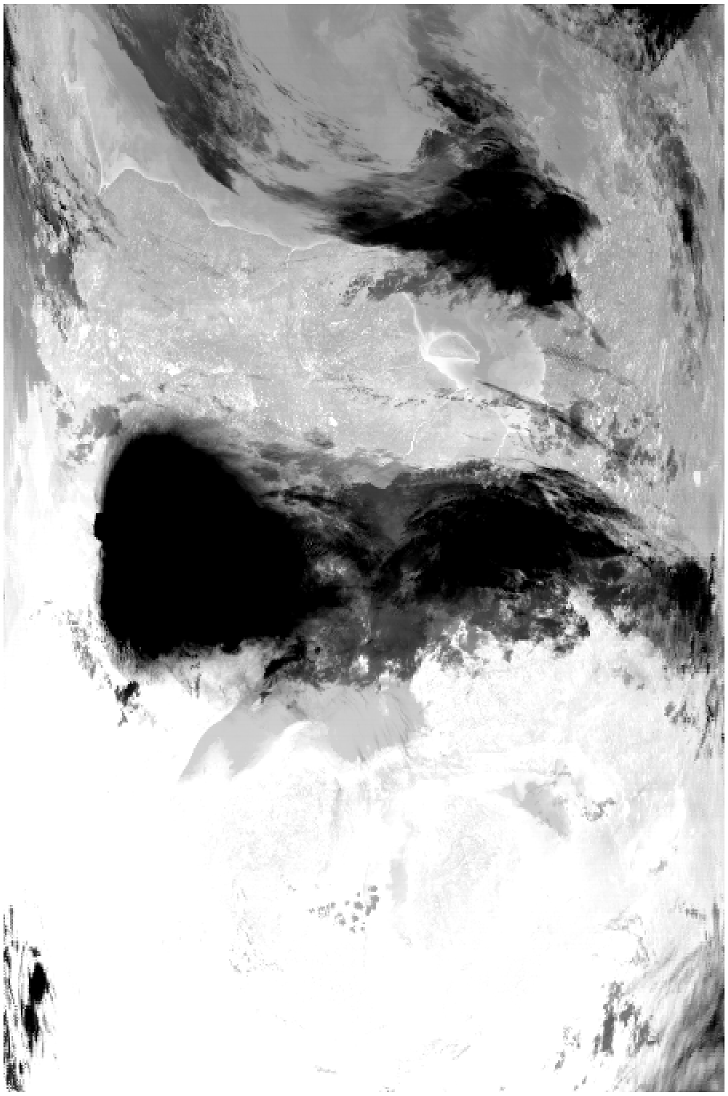


Figure 24. MWIR channel data from MYD2003183.0750 granule.

Figure 25 illustrates the cloud IP products (i.e., cloud phase, CTH, COT, and EPS) used as input in the CC/L and GCE algorithms for the MYD2003183.0750 test scene. This data includes the 16 lines per scan replication and bowtie deletion pattern. (The bowtie deletion pattern is observed to be irregular due to the scale of the image.)

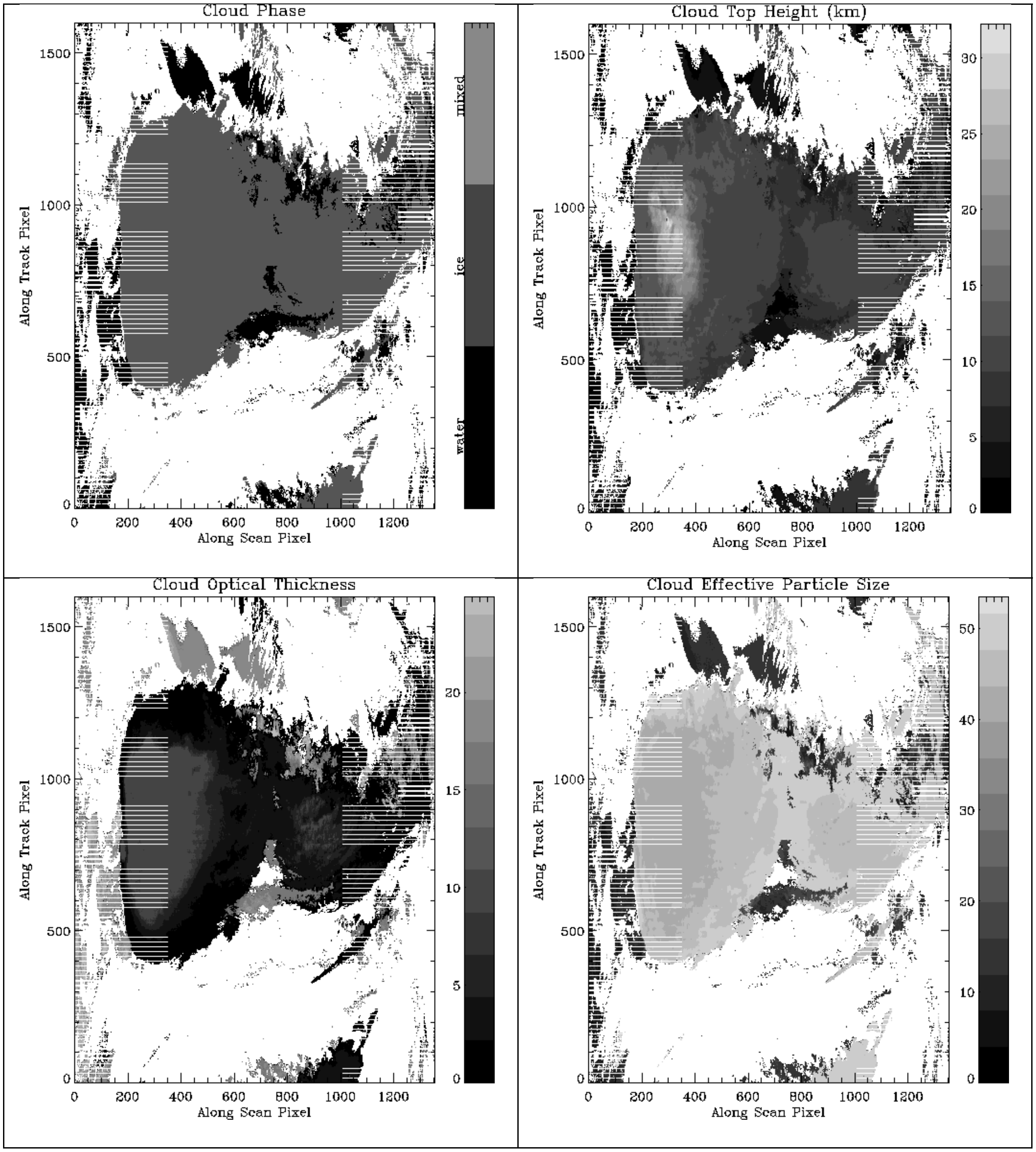


Figure 25. Cloud phase, top height, optical thickness, and effective particle size for the MYD2003183.0750 test scene.

Table 15 summarizes the cloud layer statistics produced by both the MBKM and EKM algorithms. Very little difference between the algorithms is suggested by these statistics. For example, the MBKM algorithm finds that 44.9 percent of the products cells contain a single layer of cloud, while the EKM algorithm finds 46.4 percent. The percentages of 2- and 3-layer clouds in this limited test case are smaller than might be expected globally.

Table 15. Cloud Layer Summary

	MBKM	EKM
Cloudy	50135 (51.8%)	50135 (51.8%)
Cloudy – 1 Layer	43455 (44.9%)	44886 (46.4%)
Cloudy – 2 Layers	5905 (6.1 %)	5111 (5.3%)
Cloudy – 3 Layers	766 (0.8 %)	137 (0.1%)
Cloud – 4 Layers	9 (<0.1%)	1 (<0.1%)

5.2.1. Modified Baseline K-Means Cloud Layer Algorithm

Results from the MBKM algorithm applied to the MYD2003183.0750 test scene are presented in this section. The MBKM algorithm produces cloud layer assignments that are associated with four altitude regimes (below 2.5 km, 2.5 to 5.0 km, 5.0 to 7.5 km, and above 7.5 km). The k-means clustering allows for some departures from these strict boundaries but the association between layers and the first guess height assignments remains a characteristic in the EDR products. This feature has some benefit for visualization of the results as the layers over the whole scene correspond closely to the same height range. However, some layers that straddle one of the altitude thresholds may be artificially split into two, while distinct layers wholly contained within a height regime may be identified as a single layer. Figure 26 shows the pixel-level cloud layer and type IPs produced by the MBKM algorithm.

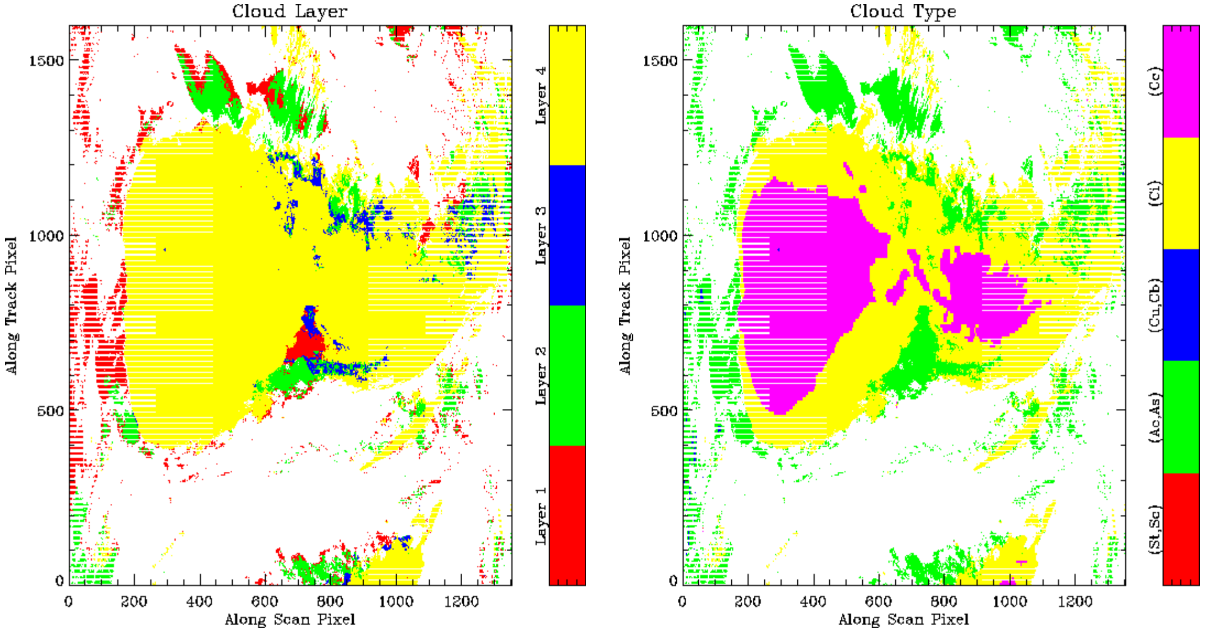


Figure 26. Pixel-level Cloud layer and type IPs for the MYD2003183.0750 test scene based on the MBKM CC/L algorithm.

The GCE results for this example are illustrated by the cloud cover, cloud type, and CTH EDRs presented in Figure 27 through Figure 31. The cloud cover EDR includes a specification of the total cloud cover over all layers (Figure 27) and the cloud cover as a function of the four layers (Figure 28). Cloud type is reported for each of the four layers (Figure 29). And cloud top height is reported as an average over all layers (Figure 30) and as a function of the four layers (Figure 31). This data represents gridded products in the sensor projection but with bowtie effects removed and with a reporting interval of approximately 5 km. (For VIIRS, the reporting interval is 6 km). The results for the other EDRs (CTT, CTP, COT, and EPS) are similar.

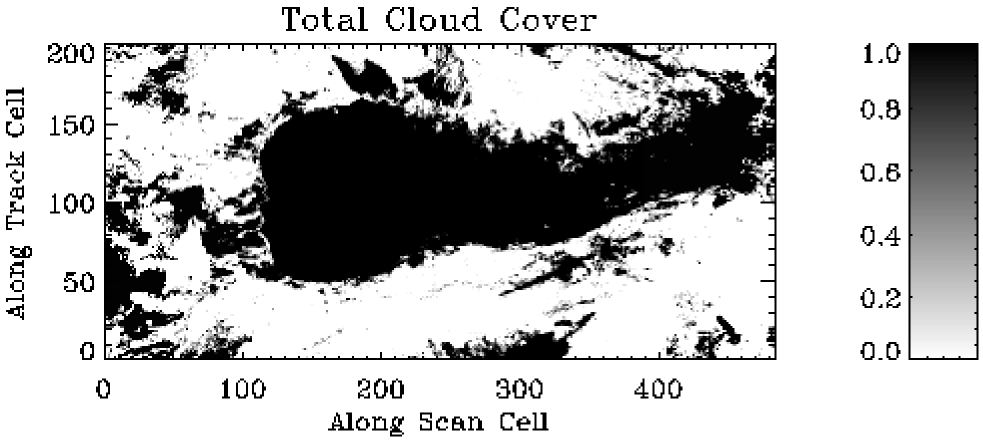


Figure 27. Cloud Cover Fraction EDR based on the MBKM CC/L layer assignments averaged over all layers.

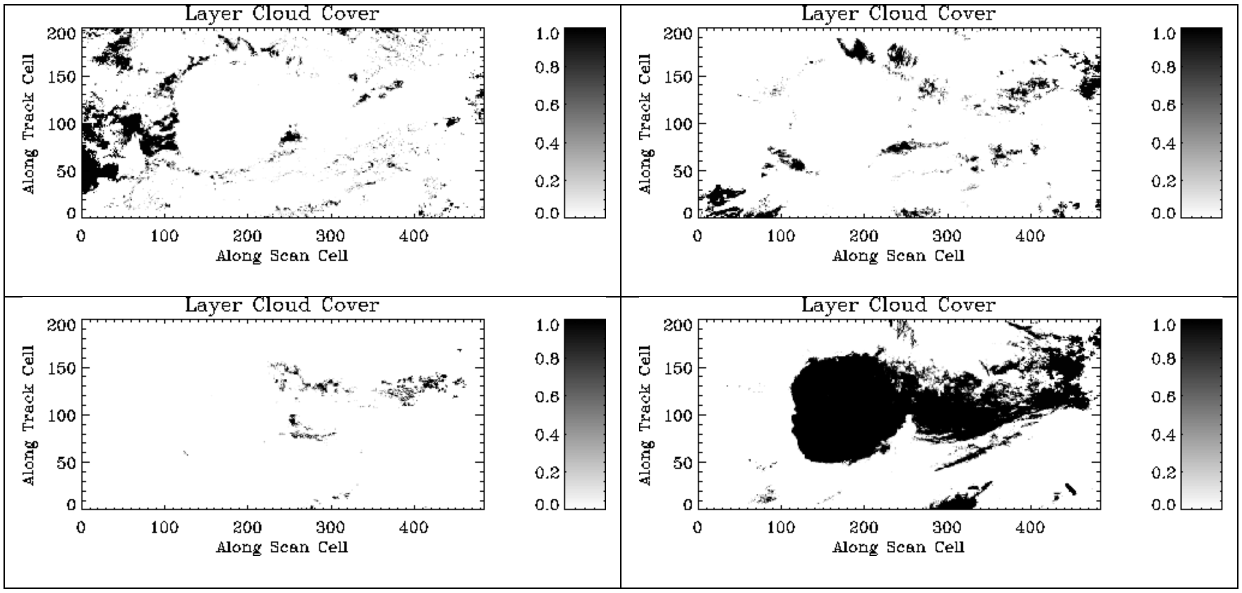


Figure 28. Cloud Cover Fraction EDR based on the MBKM CC/L layer assignments (a) for layer 1, (b) for layer 2, (c) for layer 3, (d) for layer 4.

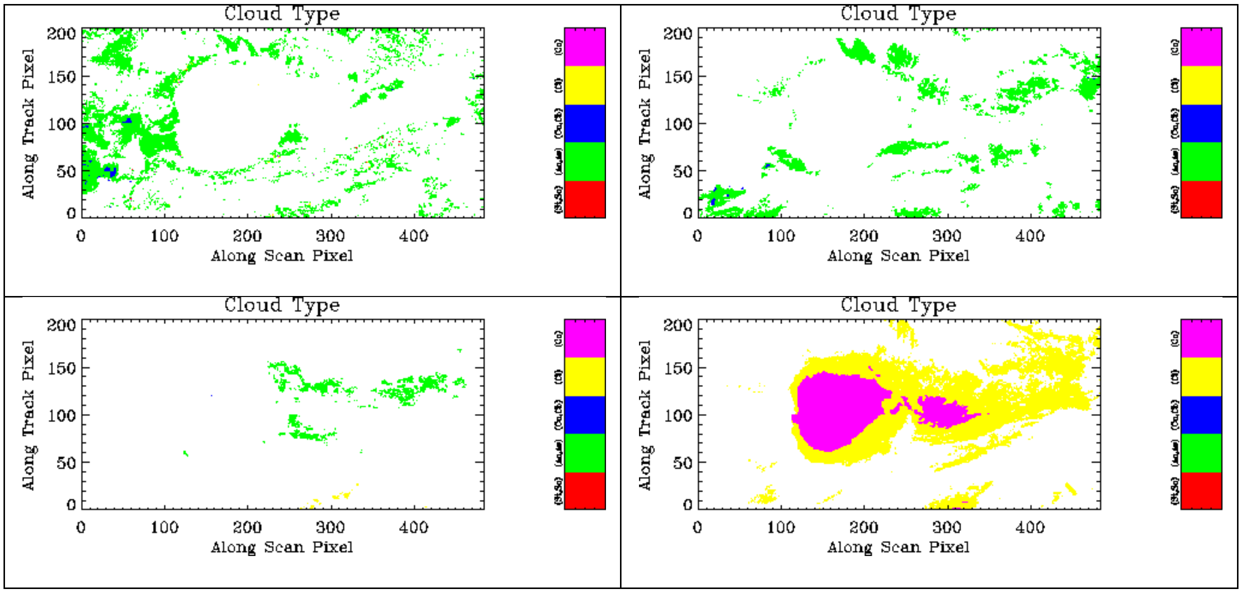


Figure 29. Cloud Cover Type EDR based on the MBKM CC/L layer assignments. (a) for layer 1, (b) for layer 2, (c) for layer 3 (d) for layer 4.

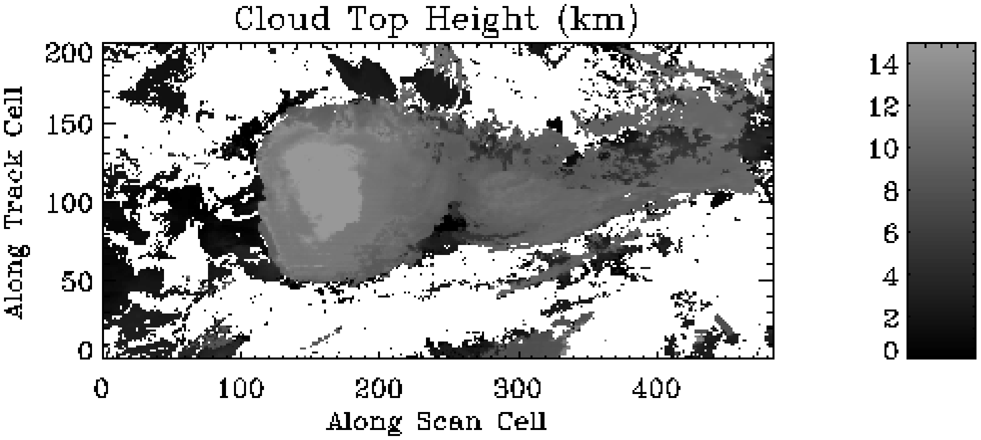


Figure 30. Cloud Top Height EDR based on the MBKM CC/L layer assignments averaged over all layers.

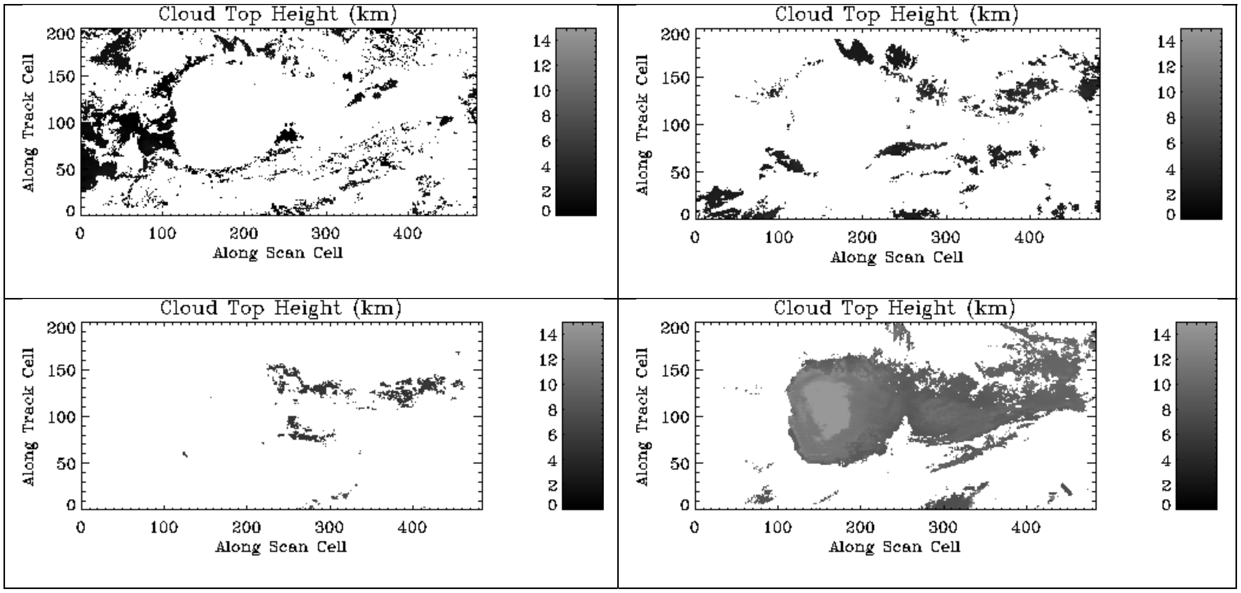


Figure 31. Cloud Top Height EDR based on the MBKM CC/L layer assignments. (a) for layer 1, (b) for layer 2, (c) for layer 3, (d) for layer 4.

5.2.2. Extended K-Means Cloud Layer Algorithm

Results from the EKM algorithm applied to the MYD2003183.0750 test scene are presented in this section. The EKM algorithm produces cloud layer assignments with relative heights. When one cloud layer is present in the horizontal reporting cell (HC), this cloud is always assigned to the first layer regardless of the cloud height. If a second cloud layer (at higher altitude) is present, this cloud is assigned to the next layer. Up to 4 layers may be identified for a given HC. The pixel-level cloud layer and type IPs for the EKM algorithm are presented in Figure 26. The appearance of the cloud layer IP differs from that of the MBKM algorithm. In this case, the IP identifies the number of layers in the vicinity of each product cell but contains no absolute indication of cloud height. On the other hand the cloud type assignments are very nearly the same as that produced by the MBKM algorithm.

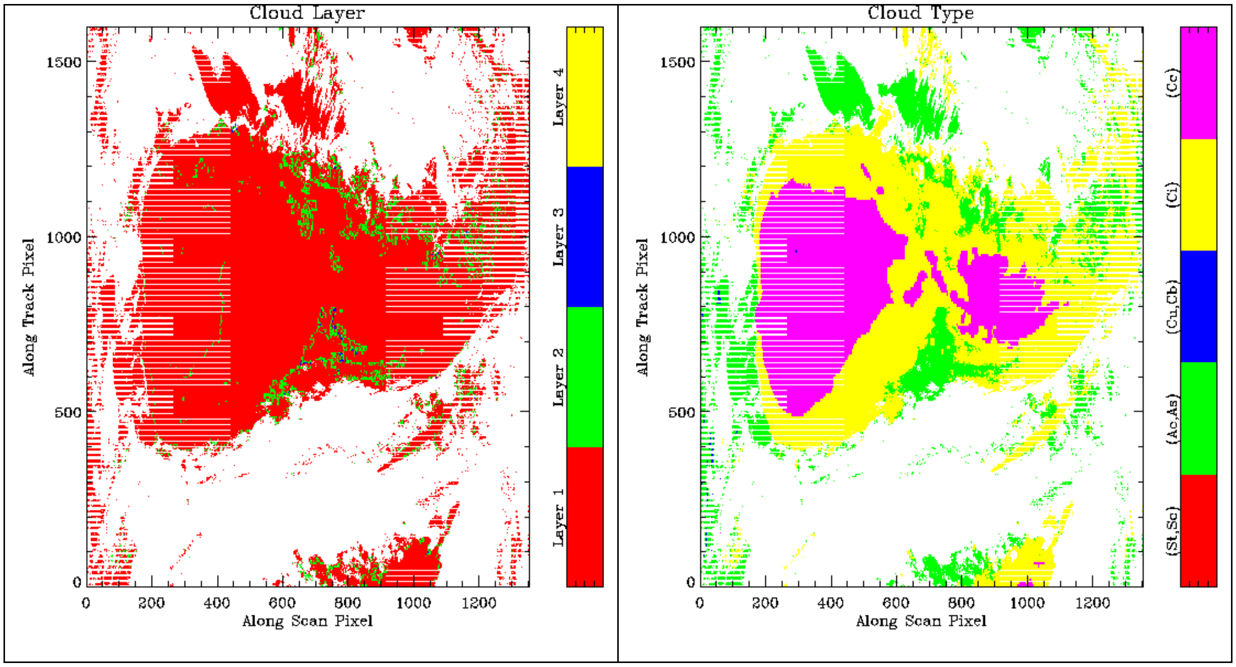


Figure 32. Pixel-level Cloud layer and type IPs for the MYD2003183.0750 test scene based on the EKM CC/L algorithm.

The GCE results for this example are illustrated by the cloud cover, cloud type, and CTH EDRs presented in Figure 33 through Figure 37. The cloud cover EDR includes a specification of the total cloud cover over all layers (Figure 33) and the cloud cover as a function of the four layers (Figure 34). Cloud type is reported for each of the four layers (Figure 35). And cloud top height is reported as an average over all layers (Figure 36) and as a function of the four layers (Figure 37). This data represents gridded products in the sensor projection but with bowtie effects removed and with a reporting interval of approximately 5 km. (For VIIRS, the reporting interval is 6 km). The results for the other EDRs (CTT, CTP, COT, and EPS) are similar.

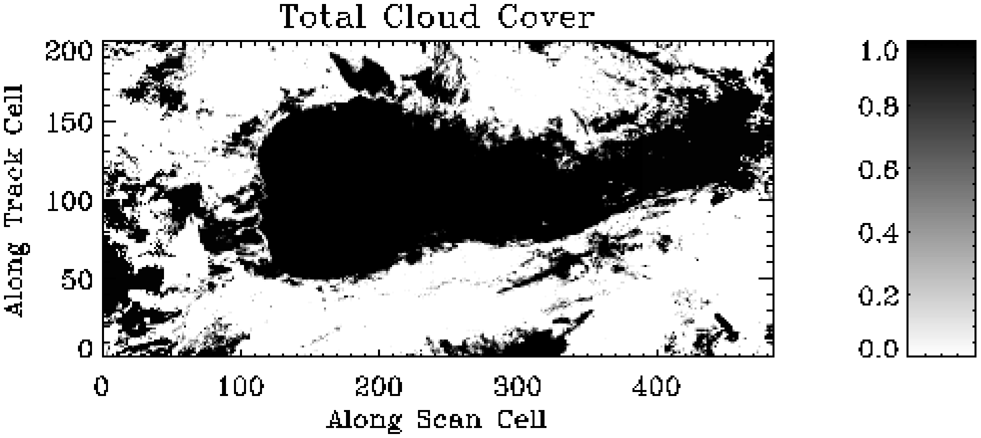


Figure 33. Cloud Cover Fraction EDR based on the EKM CC/L layer assignments averaged over all layers.

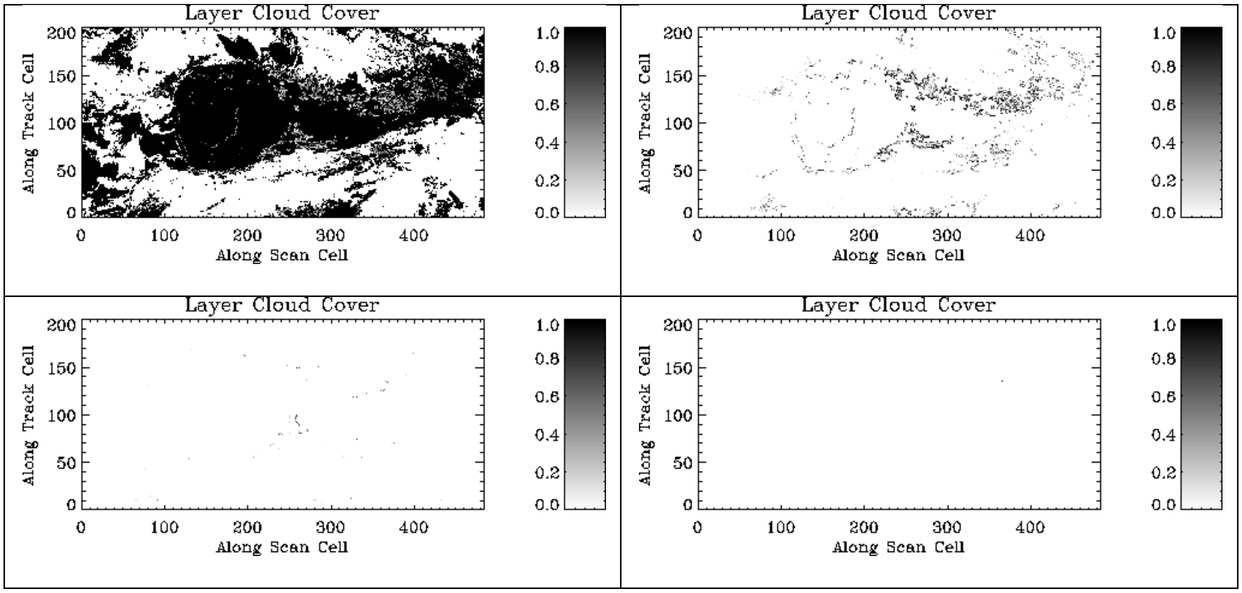


Figure 34. Cloud Cover Fraction EDR based on the EKM CC/L layer assignments (a) for layer 1, (b) for layer 2, (c) for layer 3, (d) for layer 4.

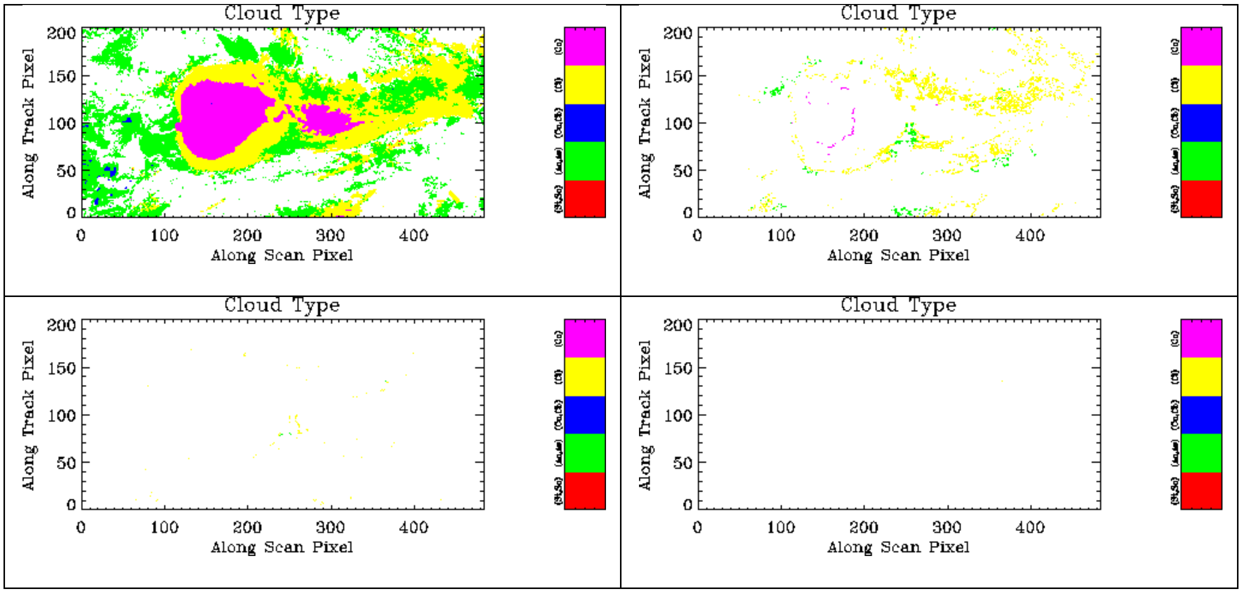


Figure 35. Cloud Cover Type EDR based on the EKM CC/L layer assignments. (a) for layer 1, (b) for layer 2, (c) for layer 3 (d) for layer 4.

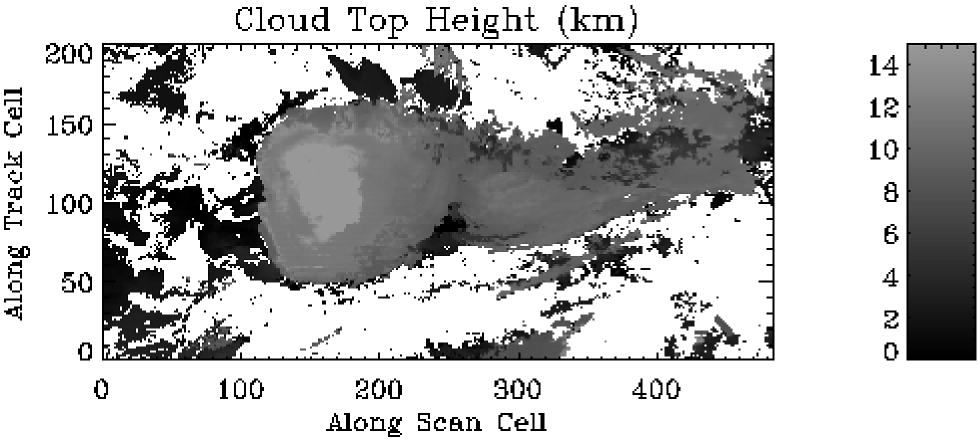


Figure 36. Cloud Top Height EDR based on the EKM CC/L layer assignments averaged over all layers.

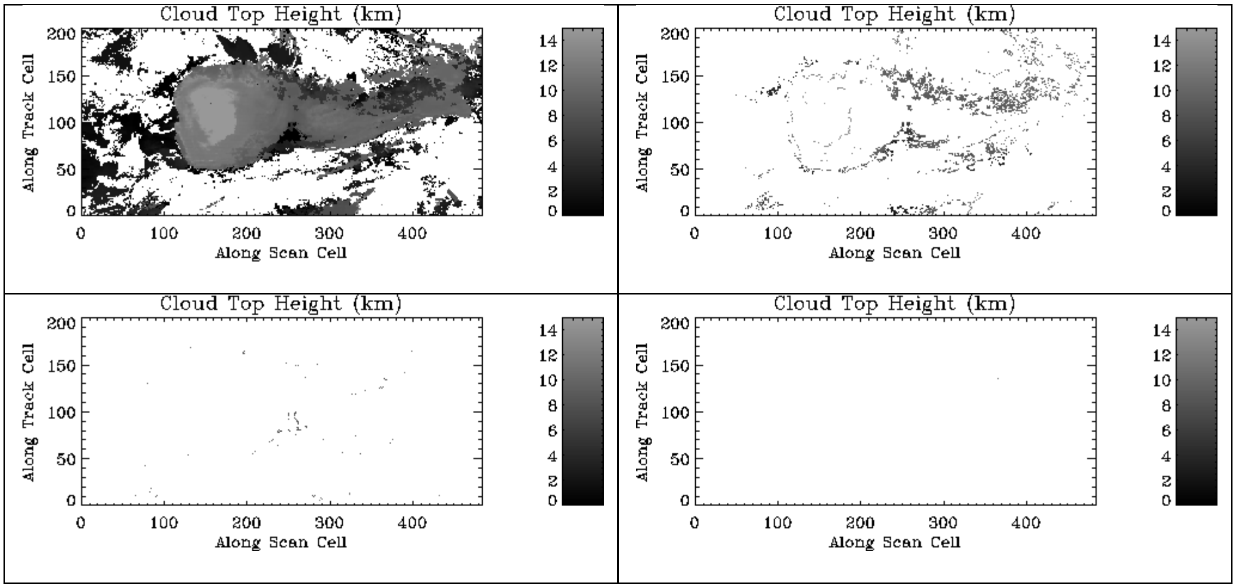


Figure 37. Cloud Top Height EDR based on the EKM CC/L layer assignments. (a) for layer 1, (b) for layer 2, (c) for layer 3, (d) for layer 4.

5.2.3. Diagnostic Analysis of the MBKM Algorithm

The results presented in the previous sections illustrate the products generated by the CC/L and GCE modules. It is clear from these illustrations that the CC/L MBKM and EKM algorithms result in cloud layer assignments with somewhat different characteristics. However, it is difficult to assess from these plots how good the cloud layer assignments are. To investigate the performance of the algorithms, a set of diagnostic plots have been created. These plot show the cloud layer and type assignments for the product cell pixels as a function of positions and cloud properties (i.e., phase, CTH, COT, EPS). With this tool it is possible to assess whether the CC/L algorithm is performing as expected. By looking a diagnostic plots for many product cells, the performance of the MBKM and EKM algorithms are evaluated in Section 5.2.5. This section presents a single example of a diagnostic plot for CC/L results produced using the MBKM algorithm.

Figure 38 shows an example of a diagnostic plot based on results from the MBKM algorithm. The top half of the plot shows results pertaining to the cloud layer assignment. The cloud layer assessment includes a key (top left) that identifies water clouds with circles, ice clouds with triangles, and mixed-phase clouds with squares. Those pixels that comprise the product cell are further identified with a layer via the color-coding: layer 1 = red, layer 2 = green, layer 3 = blue, layer 4 = magenta. A CTH image showing the pixels belonging to the cluster cell is shown to the right of the key. CTH is represented by grey shades with darker shades indicated lower altitudes, lighter shades indicating higher altitudes, and white areas representing cloud-free regions. All pixels within the product cell are assigned a colored symbol that indicated the assigned layer. To the right of this, the cloud cover fraction product is shown for each layer. The scatter plots of CTH versus COT and CTH versus EPS and the histogram of CTH presented in the next row of plots can be used to compare the algorithm layer assignments with subjective identifications. In this case, all clouds above 7.5 km are identified with a single layer, while the distributions suggests that two distinct layers may be present. This tendency to group clouds into a single bin (especially those above 7.5 km) is a limitation of the MBKM algorithm. The CTH histogram illustrates the distributions based on all pixels in the cluster cell (black line) and that for each layer in the product cell (colored lines).

The bottom half of the plot shows results pertaining to the cloud type assignment. In this case the product pixels are identified with cloud type as follows: type 1 = red, type 2 = green, type 3 = blue, type 4 = magenta, type 5 = turquoise as described by the key. Refer to Table 12 for cloud type descriptions. The plots to the right show the CTH image and type identification and the cloud type versus layer product. The bottom row shows the cloud type assignments as functions of CTH versus COT and CTH versus EPS scatter plots. The color-coded ellipses represent the range of CTH, COT, and EPS for each cloud type as specified in Table 12. Also shown is a plot of the normalized distance metric that compares the cloud properties associated with each layer and the empirical values for each cloud type. The minimum distance should correspond to the assigned cloud type.

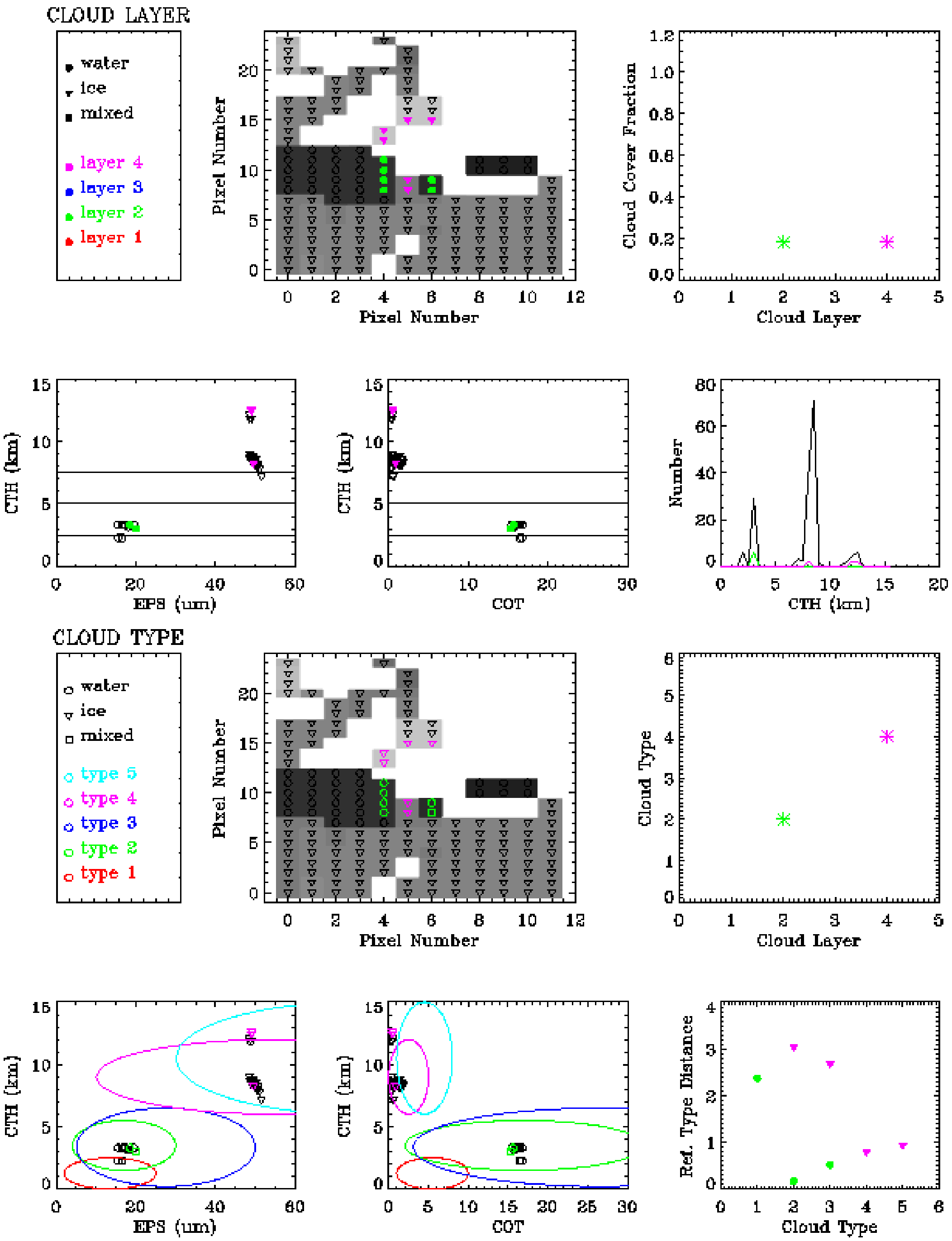


Figure 38. Diagnostic Plot based on results from the MBKM algorithm

5.2.4. Diagnostic Analysis of the EKM Algorithm

Diagnostic plots based on results produced by the EKM algorithm are presented in Figure 39. Refer to Section 5.2.3 for a description of this illustration. The main difference between the results produced by the EKM algorithm and those of the MBKM algorithm is that the EKM algorithm finds three layers while the MBKM algorithm found only two. The cloud type assignments are identical. Also the EKM algorithm orders all layers consecutively by height with the lowest layer assigned to layer 1, the next to layer 2, etc. The cloud type assignment is identical to that produced by the MBKM algorithm.

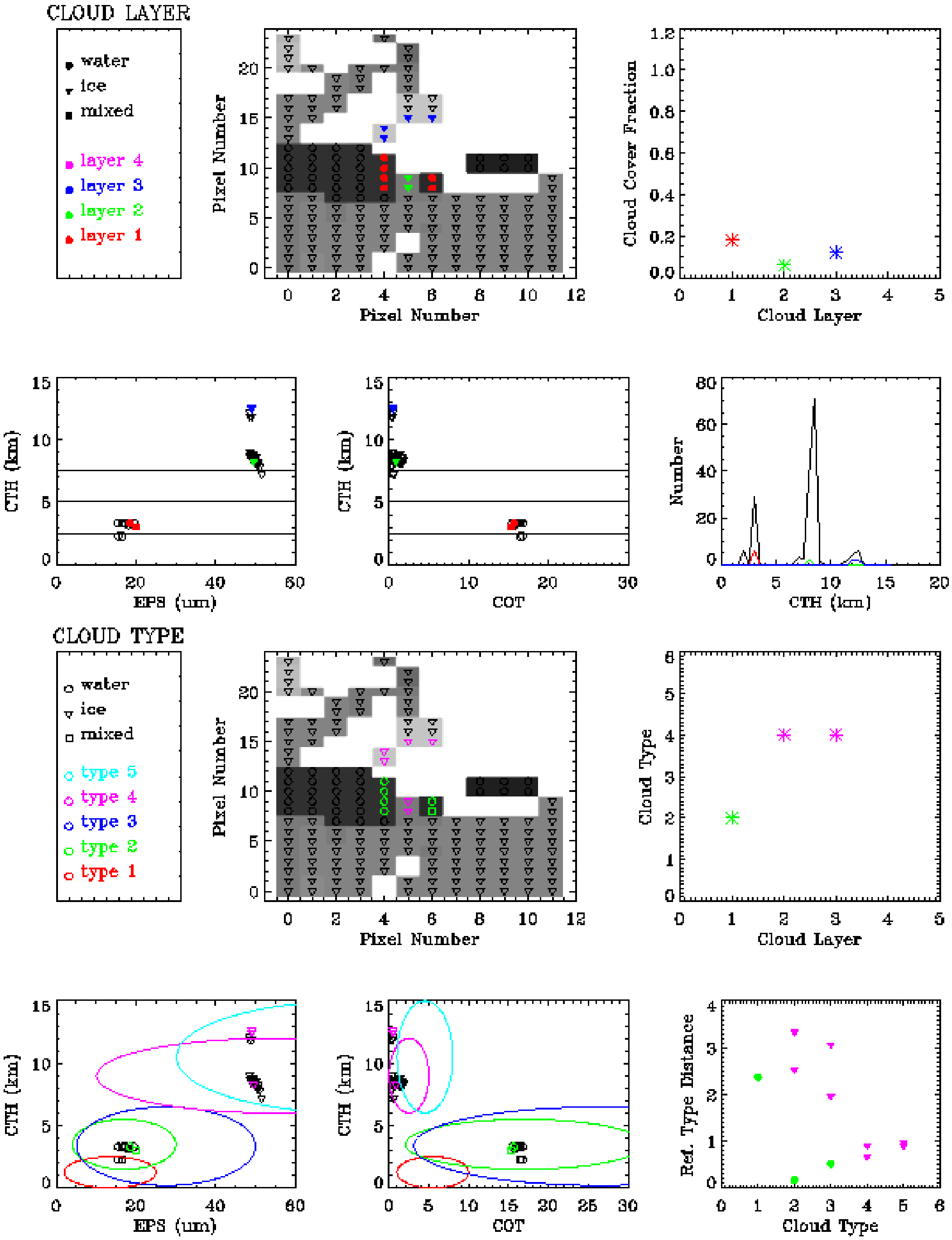


Figure 39. Diagnostic Plot based on results from the EKM algorithm

5.2.5. Algorithm Performance Evaluation

This section provides a summary of the CC/L and GCE algorithm performance based on a subjective assessment by a human analyst for a sample of 100 cells. For each cell, the performance of the automated layer assignments was evaluated subject to the following categories:

PREFORMANCE CATEGORY

A -- Good performance, same as subjective assessment

B -- Small number of pixels mis-assigned (<15%)

C -- Single layer bifurcated into 2 or more

D -- Two layers joined into one

E -- significant number of misassigned pixels (> 15%), or other major problem

This summary coupled with sample displays provides assurance as to the proper functioning of the algorithm.

The evaluations were based on cells selected via an automated process resulting in cases that included clouds in multiple layers (within the cluster cell). Table 16 shows the score sheet describing the number of layers identified N1 and score S1 for the MBKM algorithm and for the EKM algorithm (N2 and S2).

Table 16. Score Sheet for 1- MBKM and 2- EKM Performance Assessment

#	N1	S1	N2	S2	#	N1	S1	N2	S2	#	N1	S1	N2	S2	#	N1	S1	N2	S2
0	2	A	2	A	25	3	D	3	A	50	2	B	1	B	75	3	C	3	B
1	2	A	2	A	26	1	A	1	A	51	2	A	2	A	76	3	C	2	A
2	1	A	1	A	27	2	A	2	A	52	3	B	3	B	77	2	A	2	A
3	1	A	1	A	28	1	A	1	A	53	2	A	2	A	78	3	B	3	B
4	2	A	2	A	29	1	D	2	A	54	2	A	2	A	79	3	C	2	A
5	2	B	2	A	30	1	D	2	A	55	2	A	2	A	80	3	C	2	A
6	2	D	3	A	31	1	D	2	A	56	1	A	1	A	81	2	C	1	A
7	2	A	2	A	32	1	D	2	A	57	1	A	1	A	82	3	C	2	A
8	2	A	2	A	33	2	D	2	A	58	1	A	1	A	83	3	C	2	A
9	2	A	2	A	34	1	D	2	A	59	3	C	2	A	84	2	A	2	A
10	2	A	2	A	35	1	A	1	A	60	3	C	2	A	85	2	A	2	A
11	2	A	2	A	36	2	D	3	A	61	1	A	1	A	86	3	C	2	A
12	2	D	3	A	37	2	A	2	A	62	2	A	2	A	87	4	C	4	C
13	2	A	2	A	38	2	A	2	A	63	2	A	2	A	88	2	B	2	B
14	0	A	0	A	39	0	A	0	A	64	2	A	2	A	89	2	A	3	C
15	2	A	2	A	40	2	A	2	A	65	3	A	3	A	90	2	A	2	A
16	2	D	3	A	41	2	A	2	A	66	2	A	2	A	91	3	C	2	A
17	2	A	2	A	42	3	B	3	A	67	2	A	2	A	92	2	A	2	A
18	3	A	2	A	43	3	B	2	B	68	2	B	3	B	93	1	A	1	A

#	N1	S1	N2	S2	#	N1	S1	N2	S2	#	N1	S1	N2	S2	#	N1	S1	N2	S2
19	1	D	2	A	44	1	A	1	A	69	3	B	3	B	94	1	A	1	A
20	1	A	1	A	45	2	A	2	A	70	2	A	2	A	95	1	A	1	A
21	2	A	2	A	46	2	A	2	A	71	2	A	2	A	96	1	A	1	A
22	2	D	3	A	47	3	B	3	B	72	2	A	2	A	97	3	C	2	A
23	2	A	2	A	48	1	A	2	A	73	1	A	1	A	98	3	C	2	A
24	1	D	2	A	49	2	A	2	A	74	2	A	2	A	99	1	A	1	A

Figure 40 below provides a histogram of the results. The EKM algorithm provides excellent performance with ~90% of the cells having identical results to the subjective assessment and about 98% (90+8%) with no or few (<15% differences) in pixel assignments. The MBKM Algorithm has ~60% of the cells classified identically to the subjective assessment and about 70% with no or few differences. As can be seen the differences primarily result from the MBKM's significant occurrences (15% each) of Category C (single layer per subjective assessment split by algorithm into two or more layers) and Category D (merging of two different into one by the algorithm). These are a consequence of the high dependence of the final result on the first guess assignments in the MBKM Algorithm.

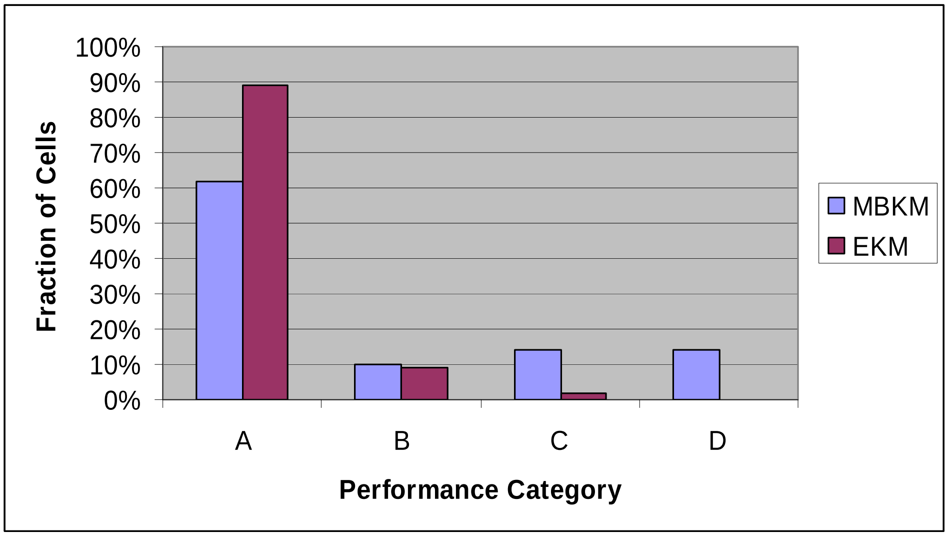


Figure 40. Comparison of MBKM and EKM Algorithm Layer Classifications to Subjective Assessment for 100 Horizontal Cells

For the cases considered, the score for the EKM algorithm was higher than the MBKM algorithm. The poorer performance of the MBKM algorithm was largely attributed the following two problems.

- D-Rating: Two distinct layers at high altitude (> 7.5 km) were not identified as separate layers (Cell number 6, 12, 16, 19, 22, 24, 25, 29, 30, 31, 32, 33, 34, and 36).

- C-Rating: Single layers located at 2.5 km (Cell number 75, 76, 79, 80, 81, and 82) and at 5.0 km (Cell number 59, 60, 83, 86, 87, 91, 97, and 98) were incorrectly identified as two layers.

The layer classifications were given a B-rating (a few misidentified pixels) when there were two closely spaced layers with some scatter in CTH between the layers making a clear separation into layers difficult. Both algorithms performed comparably under these conditions. The assignments in these cases are largely dependent on the weighting given to the other k-means parameters (i.e., EPS, COT, and phase).

Only one case (87) was identified with 4 layers (by both algorithms). A diagnostic plot of this case is presented in Figure 41. Inspection of this case, suggests that only three layers are present (at 1.5, 5.5, and 10 km). Some points with CTH between the two lower layers have been incorrectly identified as a fourth layer. These pixels likely represent fields of view that include signal from the two lower layers (i.e., multi-layer cloud).

The EKM algorithm under-performed relative to the MBKM algorithm (receiving a C-rating) in one instance (89). In this case, two closely spaced layers were identified but the weighting given to CTH and EPS in the k-means analysis resulted in layer reassignments that disagree with the subjective evaluation.

CDFS-II has a greater frequency of occurrence of three or four layers than is observed with either the MBKM or EKM Algorithms. Typical CDFS-II results identify 40% of cells with two layers 15% with three layers and ~1% have four layers (Gustafson, 2005). We believe this is due mostly to the significantly smaller HCS of the algorithm in this document compared to CDFS-II. The test data here had cells of 5 km. The CDFS-II results are based on cells of ~25 km.

In conclusion, the EKM algorithm corresponds most closely to a subjective assessment by a human analyst with 90% of the cells classified identically and 98% with no or only a few differences). The MBKM Algorithm provides reasonable performance (60% identical to subjective assessment and about 70% with no or few differences).

Based on these results we tend to recommend that the EKM be selected. However, some users might prefer the quasi-fixed layer assignments of the MBKM algorithm. For this reason we have delivered the algorithm software with both algorithms (selectable by a configuration parameter).

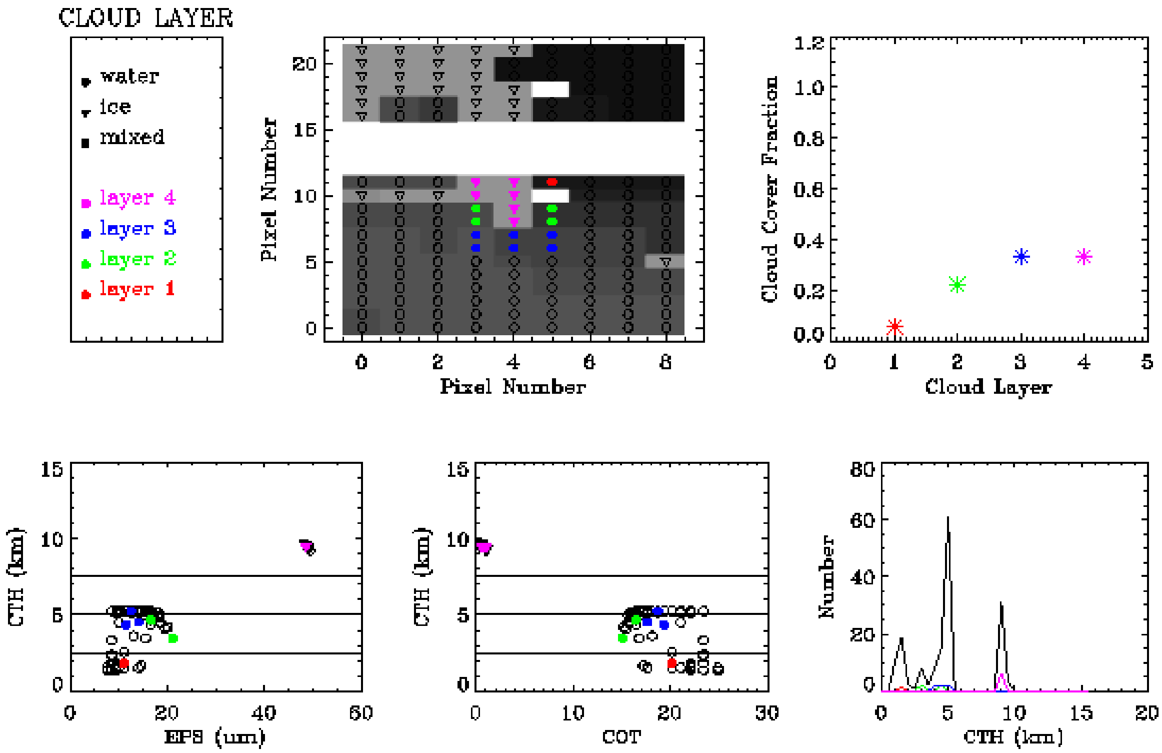


Figure 41. Product cell with 4 layers identified. The EKM and MBKM routines give the same result. Inspection of the data suggests 3 layers are present and that some pixels may represent multi-layer cloud.

5.3. PRACTICAL CONSIDERATIONS

Paragraph SRDV3.2.1.5.4-1 of the VIIRS SRD states the following:

> "The scientific SDR and EDR algorithms delivered by the VIIRS contractor shall be convertible into operational code that is compatible with a 20 minute maximum processing time at either the DoD Centrals or DoD field terminals for the conversion of all pertinent RDRs into all required EDRs for the site or terminal, including those based wholly or in part on data from other sensor suites."

RDR here stands for Raw Data Record. This essentially means that any and all EDRs must be completely processed from VIIRS raw data, including calibration and geo-referencing within 20 minutes from the time the raw data are available. This requirement is a strong reminder that VIIRS is an operational instrument.

In the CC/L algorithm, numerical approximations to various statistical tests are performed. None of these, however, have unstable numerical properties nor are they applied in an iterative fashion, so as to amplify numerical errors. Therefore, the CC/L algorithm is resistant to numerical problems associated with finite precision arithmetic on computers. CC/L is designed to trade accuracy with computing resources. If the initial CTH/Phase does not provide accurate enough starting partitions, the Property Distance Cluster K-Means algorithm iterates to improve layer and cloud typing accuracy.

5.4. ALGORITHM VALIDATION

The CC/L algorithm will be validated against MAS data and MODIS data along the imagery track centers where lidar cloud profiling data are available. By inspection, a set of ground truth layered cloud amounts will be determined. These will then be compared to CC/L layered assessments and a qualitative indication of CC/L performance can be attained. At least one case of MODIS data will be used to validate CC/L algorithm performance. The EDR will be validated with independent cloud measurements either from space or other indirect means. The required validation data and procedure that can be used for validating algorithm performance can be briefly summarized as:

- Collect statistically significant samples of co-located in-situ cloud layer/type measurements and VIIRS-like measurements.

- Modify/create VIIRS-like measurements with VIIRS instrument specification noise.

- Perform EDR retrieval using ATBD described algorithms.

- Co-register in-situ data and EDR retrievals by taking into account spatial, temporal, and viewing discrepancy.

- Perform statistical accuracy, precision, and uncertainty estimates of EDR using retrievals and in-situ data.

6. ASSUMPTIONS AND LIMITATIONS

6.1. ASSUMPTIONS

6.1.1. Perform Parallax Correction method

The aggregation method assumes that the perform parallax correction method produces an output where the IP values are at the same locations (latitudes and longitude) as the observed pixels (i.e., geolocation given in the SDR file). That is, the latitudes and longitude grid remains unchanged and the IP values are moved as necessary due to the parallax effect.

6.1.2. Processing Note – Edge Effects

VIIRS data will be processed in groups of scans called granules. Each such granule must of necessity have a start (first scan and stop) last scan. As shown in Figure 20, the clustering cells (although not the product cells) in the first and last scans will spill over the region of available data. From a software perspective, this condition is easily handled by simply ignoring the missing pixels and this feature is implemented in the algorithm. A preferred method use data from the previous granules to take care of the first scan in a granule and to wait until the next scan is available (this might entail waiting for the next satellite contact). This is illustrated in Figure 42 which shows 10 scans covering three granules (the numbering is arbitrary). The current granule or contact covers scans #s 136 through 141, the previous scan ends at scan #135 and the next begins at scan # 142.

The processing strategy as indicated in Figure 42 is to prefix input data (the various IPs used for CC/L) from the last two scans of the previous granule (#s 134, 135) to the before data from scans of the current granule (or contact). In the example shown, the algorithm then derives EDRs only for scan #s 135 through 140, i.e., all but the first and last scan of the input data. When the next granule is processed, IPs for scan #s 140 and 141 would be prefixed to the granule and processing would occur similarly. The CC/L layering would be output for granule 141, the one not provided in the earlier processing cycle.

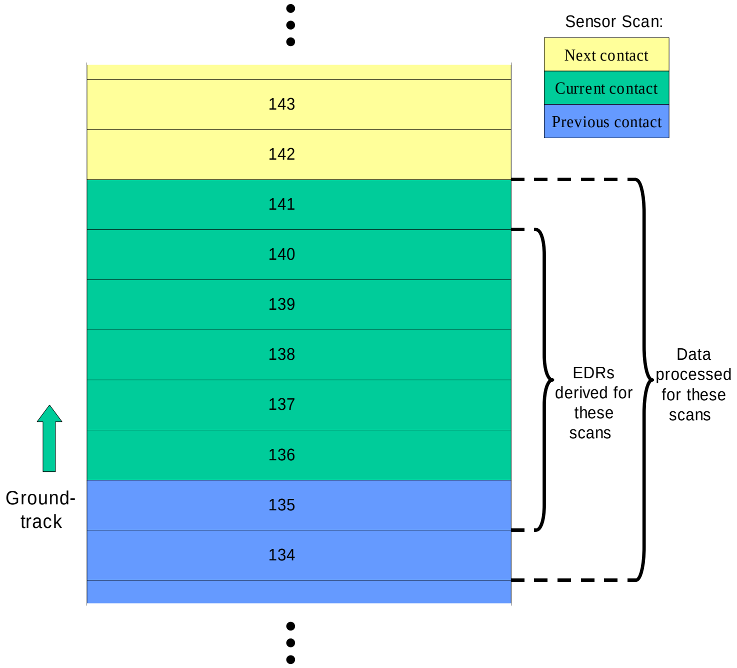


Figure 42. Processing Approach to Handle Edge Effects. Scans are identified by an arbitrary sequential number. (Note: This is a logical view – geometric overlap and bow-tie deletion are not indicated on this figure)

This approach provides overlap of the data used to compute the clusters between granule and will reduce any potential discontinuities in layer assignments between granules.

Our software is designed to support both options based on a configuration setting:

- “process all” → process all data provided including first and last scans (ignoring any clustering cell pixels that spill over to other granules)
- “ignore first and last” → do not process the first and last scans, no clustering cell pixels need be ignored in this case

If the latter (“ignore first and last”) option is selected it is the responsibility of the algorithm processing infrastructure to prefix the additional two scans of data to the input data. We note the first option might be preferred for a tactical terminal, when it might not be possible to easily acquire data from scans before or after a contact.

Since the CC/L algorithm defines the layering assignments for the computation of all the output cloud EDRs as produced in the GCE Unit, the GCE inputs must also be configured identically to those used to produce CC/L. That is, when “ignore first and last” is set, all the cloud IPs and other support data must for the same identical regions as used in the CC/L (i.e., they must also have IPs from the last two scan of the previous granule prefixed to the input data). This is also the responsibility of the algorithm processing infrastructure.

6.1.3. Cloud Type Definition

Cloud typing definition is based on current general knowledge of cloud microphysical properties. Definition of cloud type is subject to being updated when newer and more sophisticated cloud property measurements can reveal any improved cloud type information.

6.1.4. Scan Angle Correction

The clouds, considered for valid scan angle correction, are assumed to be totally and uniformly single layer opaque cumulus cloud.

6.2. LIMITATIONS

6.2.1. Inherent Cluster Ambiguity

There is no best mathematical classification of data into clusters. Clustering is a fundamentally soft subject, where many different and equally justifiable interpretations of data are present. Any single partition of a dataset is inherently ambiguous. Rescaling, rotations, arbitrary coordinate transforms, choices of distance metrics, selection of the cluster integrity measures, and other design decisions determine the performance of an algorithm.

6.2.2. Definition of Cloud Layers/Types

Various definitions of cloud layers/types are defined and discussed. The choice of cloud layer definition dramatically affects the design and implementation of a cloud layer algorithm. The algorithm inputs ultimately limit the clustering capability.

6.2.3. Cumulus Cloud Model for Cloud Fraction Angle Correction

The baseline VIIRS cloud cover viewing angle correction approach and assumption are discussed above. A single layer of opaque cloud is assumed, and only the single variable of cloud masking exponent is modeled to account for complex cloud properties such as size, shape, thickness, spacing and height. The viewing angle dependent resolution of the VIIRS field of view is only considered through empirical first order adjustment. Multiple-layer clouds are not addressed.

7. REFERENCES

Arking A., and J. D. Childs, 1985: Retrieval of Cloud Cover Parameters from Multispectral Satellite Images. Journal of Climate and Applied Meteorology, 24, 322-333.

AER, 2004: VIIRS Cloud Cover/ Layers and Generate Cloud EDRs, Technical Interchange Meeting, 9 November 2004.

Burrough, P. A., 1986: Principles of Geographical Information Systems for Land Resources Assessment. Clarendon Press.

Desbois, M., G. Seze and Szejwach, 1982: Automatic classification of clouds on METEOSAT imagery: Application to high-level clouds. Journal of Applied Meteorology, 21, 401-412.

Dowling, D.R., and L.F. Radke, 1990: A summary of the physical properties of cirrus clouds, J. Appl. Meteor., 29, 970-978.

ECS Project, 1995: "Theoretical Basis of the SDP Toolkit Geolocation Package for the ECS Project", Technical Paper, 445-TP-002-002, prepared under contract NAS5-60000, P. D. Noerdlinger, Hughes Applied Information Systems, Landover, Maryland, May 1995

Fye, F. K., 1978: The AFGWC automated real-time cloud analysis model, AFGWC Tech. Note 78/002, Air Force Global Weather Central, Omaha, NE, 97 pp.

Gustafson, Gary, 2005, personal communication.

Hamill, T. M., d'Entremont, R. P., and J. T. Bunting, 1992: A description of the Air Force Real-Time Nephanalysis Model, Weather and Forecasting, 7, 288-306.

Heymsfield, A.J., and C.M.R. Platt, 1984: A parameterization of particle size spectrum of ice clouds in terms of the ambient temperature and ice water content. J. Atmos. Sci., 41, 846-855.

Liou, K.N., 1992: Radiation and cloud processes in the atmosphere, Oxford University Press, New York, pp.487.

MacQueen, J., 1967: Some methods for classification and analysis of multivariate observations. In Le Cam, L. M. and Neyman, J., editors, Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume I: Statistics, pages 281-297. University of California Press, Berkeley and Los Angeles, CA.

Minnis, P., Viewing Zenith Angle Dependence of Cloudiness Determined from Coincident GOES East and GOES West Data, J. Geophys. Res., 94, 2303-2320, 1989.

Mohoney, M. J., 2004, "A Discussion of Various Measures of Altitude," http://mtp.jpl.nasa.gov.notes/altitude/altitude.html (October 22, 2004)

NIMA, 1997, NIMA TR8350.2: “Department of Defense World Geodetic System 1984 – Its Definition and Relationship with Local Geodetic Systems”, Third Edition, prepared by the NIMA WGS84 Development Committee.

Planet, W.G. (ed.), 1988: Data extraction and calibration of TIROS-N/NOAA radiometers. NOAA Technical Memorandum NESS 107 – Rev. 1, Oct. 1988. 130 pp.

Selim S.Z., and M A Ismail, 1984: "K-means-type algorithms: A generalized convergence theorem and characterisation of local optimality", IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 1, pp. 81-87.

Snow, J.W., Field of Cumuliform Clouds East of Cape Canaveral, Florida, EOS Trans., AGU, 67, 595, 1986.

Snow, J.W., Modeling the Variation of Cloud Cover with View Angle Using Space Shuttle Cloud Imagery, GL-TR-90-0130, Geophysics Lab report, 49pp, 1990.

Theiler J., and G. Gisler, 1997: A contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation, Proc SPIE 3159, 108-118.

Weickmann H.K., and H.J. Aufm Kampe, 1953: Physical properties of cumulus clouds. J. Meteor., 10, 204-211.