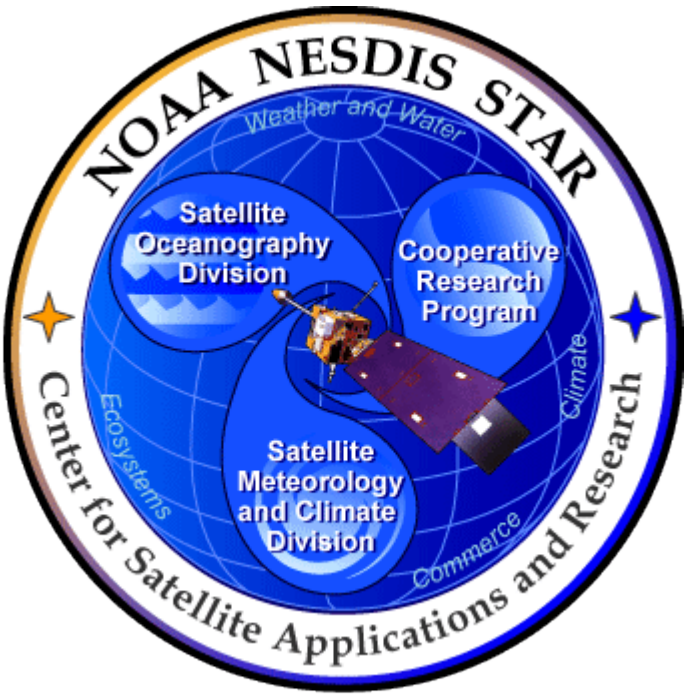


NOAA NESDIS
CENTER for SATELLITE APPLICATIONS and RESEARCH

ALGORITHM THEORETICAL BASIS DOCUMENT

Snow Cover

Andrew Rost (Lead) NOAA/NWS/NOHRSC
Thomas Painter, UCAR/UCLA
Kelley Eicher, UCAR/NOHRSC

Version 2.2

January 24, 2012

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

ABI	**A**dvanced **B**aseline **I**mager
ACM	**A**BI **C**loud **M**ask
AIT	**A**lgorithm **I**ntegration **T**eam
ATBD	**A**lgorithm **T**heoretical **B**asis **D**ocument
AVHRR	**A**dvanced **V**ery **H**igh **R**esolution **R**adiometer
AVIRIS	**A**irborne **V**isible/**I**nfrared **I**maging **S**pectrometer
AWG	**A**lgorithm **W**orking **G**roup
F&PS	**F**unction and **P**erformance **S**pecification
FSC	**F**ractional **S**now **C**over
GOES-R	**G**eostationary **O**perational **E**nvironmental **S**atellite, **R** series
GOESRSCAG	**GOES-R S**now **c**over **a**nd **g**rain size
MODIS	**Mod**erate-resolution **I**maging **S**pectrometer
MRD	**M**ission **R**equirements **D**ocument
NEDT	**N**oise **E**quivalent **D**elta **T**emperature
NDVI	**N**ormalized **D**ifference **V**egetation **I**ndex
NOAA	**N**ational **O**ceanic and **A**tmospheric **A**dministration
NWS	**N**ational **W**eather **S**ervice
QA	**Q**uality **A**ssurance
RMSE	**R**oot **M**ean **S**quared **E**rror
RTM	**R**adiative **T**ransfer **M**odel
SCAG	**S**now **c**over **a**nd **g**rain size
SNR	**S**ignal-to-**n**oise **R**atio
TM	**T**hematic **M**apper

ABSTRACT

This Algorithm Theoretical Basis Document (ATBD) provides a high-level description of the physical/mathematical basis and operational implementation of the Snow Cover (FSC) product from the Advanced Baseline Imager (ABI) to be flown onboard NOAA Geostationary Environmental Operational Satellite R series (GOES-R). Currently, prior to launch of GOES-R, the FSC algorithm is being prototyped with available satellite data, specifically NASA Moderate Resolution Imaging Spectroradiometer (MODIS) due to its spectral range, spectral sampling, and spatial resolution. The FSC algorithm, GOESRSCAG – GOES-R Snow Covered Area and Grain size, requires as its input reflectance from optical channels, brightness temperature from thermal infrared channels, and observational/illumination geometry. FSC uses a coupled multiple endmember spectral mixture analysis (MESMA) with a radiative transfer model of snow's spectral reflectance (DISORT) to estimate fractional snow cover per pixel and grain size/snow albedo of that fractional snow cover. It also estimates the fractional cover of green vegetation and soil and rock. Spectral libraries of snow account for changes in grain size, solar geometry, and view geometry. The combination of the geographically meaningful determination of fractional cover from MESMA and directionally explicit snow spectral endmembers results in a direct, physical retrieval of fractional snow cover as opposed to previous empirical approaches. The validation and testing of the FSC algorithm will be carried out with (a) retrievals of FSC from proxy ABI data (five bands) compared with retrievals of FSC for the same data but with the full band space of MODIS (seven bands – MODIS Snow Covered Area and Grain size model - MODSCAG) and with (b) comparisons of FSC from proxy ABI data with high spatial resolution retrievals of FSC from Thematic Mapper data. FSC will also be monitored with *in situ* determinations of snow presence from broad networks in the CONUS, Canada, and Alaska. Ultimately, before GOES-R ABI data are available but after the NASA MODIS on Terra and/or Aqua have failed, we will use fractional snow cover retrievals from the NOAA National Polar Orbiting Environmental Satellite System (NPOESS) analogue instrument Visible Infrared Imaging Radiometer Suite (VIIRS). Compared with MODSCAG retrievals that have an uncertainty of 0.05, the FSC from proxy ABI data have no bias (mean difference = 0.02) and a one-sigma standard deviation of 0.08 in snow cover across the range 0.00 to 1.00. These results of prototyping and validation of the ABI FSC product show that its accuracy/precision are well within existing GOES-R ABI specifications for FSC.

1. INTRODUCTION

1.1 Purpose of This Document

The Fractional Snow Cover (FSC) Algorithm Theoretical Basis Document (ATBD) provides a) a high level description of and b) the physical basis for the retrieval of the fraction of each pixel covered by snow from image data acquired by the Advanced Baseline Imager (ABI) instrument proposed for the GOES-R series of NOAA geostationary meteorological satellites. The FSC product will provide estimates of snow cover, as a fraction of each ABI pixel area, for image regions not obscured by clouds or heavy forest cover. The FSC product will be made available to variety of Algorithm Working Group (AWG) products that have indicated a dependency on *a priori* knowledge of the presence of snow. FSC, a GOES-R program office option 1 (baseline) product, has also been identified as a critical GOES-R end user product.

1.2 Who Should Use This Document

The intended users of this document are those interested in understanding the physical and mathematical basis of the spectral mixture analysis, as applied to fractional snow cover retrievals, and how to utilize the outputs of fractional snow cover for a particular application. This document also provides information useful to anyone involved with maintaining or modifying the original algorithm.

1.3 Inside Each Section

This document is broken down into the following main sections:

- **Observing System Overview**: Provides relevant details of the GOES-R ABI instrument and provides a brief description of the products generated by the Fractional Snow Cover algorithm (FSC).

- **Algorithm Description**: Provides a detailed description of the FSC including its physical and mathematical basis, its inputs and its outputs.

- **Test Data Sets and Validation**: Provides a description of the proxy GOES-R ABI data sets used to assess the performance of the FSC and the quality of its output products. It also describes the results from the FSC processing using the simulated GOES-R ABI input data.

- **Practical Considerations**: Provides an overview of the issues involved in the FSC numerical computation, programming and procedures, quality assessment and diagnostics, exception handling, and continuing validation efforts.

- **Assumptions and Limitations**: Provides an overview of the current limitations of the approach and presents a plan for overcoming these limitations with further algorithm development.

1.4 Related Documents

This document currently relates to one other document outside of the specifications of the GOES-R Ground Segment Functional and Performance Specification (F&PS), the GOES-R Mission Requirements Document (MRD) 3 and to the specific documents referenced in following sections – namely, the NOAA NESDIS Center for Satellite Applications Research Software Documentation: Fractional Snow Cover. It documents the offline version of the fractional snow cover algorithm (scag.c) as delivered to the GOES-R program office prior to the technical readiness review.

We anticipate that the FSC ATBD may ultimately relate to other GOES-R AWG ATBDs, especially the ACM ATBD.

This document includes several appendices which contain additional information for programmers:

- Appendix 1: Common Ancillary Data Sets
- Appendix 2: Description of Modified Gram-Schmidt Orthogonalization
- Appendix 3: Software Development Documentation
- Appendix 4: GOES-R Surface Reflectance Algorithm Theoretical Basis Document
- Appendix 5: Common Ancillary Data Sets for Surface Albedo and Surface Reflectance

1.5 Revision History

Version 0.1 of this document was created by Donald Cline, NOAA/NWS/NOHRSC; Thomas Painter, UCAR/UCLA; Milan Allen, NOAA/NWS/NOHRSC; Christopher Bovitz, UCAR/NOHRSC; Kelley Eicher, UCAR/NOHRSC; and Andrew Rost, NOAA/NWS/NOHRSC. Its intent is to accompany the delivery of version 0.1 of the FSC code set to the GOES-R AWG Algorithm Integration Team (AIT).

Version 1.0 of this document was created by Thomas H. Painter, UCAR/UCLA; Andrew Rost, NOAA/NWS/NOHRSC; Donald Cline, NOAA/NWS/NOHRSC. Its intent is to accompany the delivery of version 3 of the FSC code sent to the GOES-R AWG Algorithm Integration Team (AIT) and it is considered the 80% delivery. Revision date is 10 August, 2009.

Version 2.0 of this document was created by Thomas H. Painter, UCAR/UCLA; Andrew Rost, NOAA/NWS/NOHRSC, and Christopher Bovitz, UCAR/NOHRSC. Its intent is to accompany the delivery of version 5 of the FSC code sent to the GOES-R AWG Algorithm Integration Team (AIT) and is it considered the 100% delivery. Revision date is 30 June 2010.

Version 2.1 of this document was created by Thomas H. Painter, UCAR/UCLA; Andrew Rost, NOAA/NWS/NOHRSC, and Christopher Bovitz, UCAR/NOHRSC. Its intent is to accompany the September 2010 of version 5 of the FSC code sent to the GOES-R AWG Algorithm Integration Team (AIT) and is it considered an update to the 100% delivery. Revision date is 30 September 2010.

Version 2.2 of this document was created by Thomas H. Painter, UCAR/UCLA; Andrew Rost, NOAA/NWS/NOHRSC, and Kelley Eicher, UCAR/NOHRSC. Its intent is to (a) address specific questions posed by the AER; (b) provide a description of Gram-Schmidt

orthogonalization; (c) provide software development documentation; and (d) provide the algorithm theoretical basis documentation for the GOES-R surface reflectance products, and it is considered an update to the 100% delivery. Revision date is 24 January 2012.

All revisions to this ATBD include author of the revision, description of the revision, motivation for the revision, and revision number and date.

2. OBSERVING SYSTEM OVERVIEW

This section will describe the products generated by the FSC and the requirements it places upon the GOES-R ABI instrument. Where appropriate, throughout the remainder of this document, the FSC may also be referred to as GOESRSCAG (GOES-R Snow Cover And Grain size), which is the formal name of the FSC code set.

2.1 Products Generated

The FSC is responsible for calculating subpixel estimates of snow cover. FSC retrievals are expressed as the fraction of the ABI pixel covered by snow (0.0 = no snow cover continuously through 1.0 = total snow cover). FSC is often referred to in the literature as sub-pixel snow cover. Since the FSC is based upon the spectral reflectance of snow in the visible and near-visible wavelengths of the electromagnetic radiation spectrum that varies by the changes in grain size of the snow pack's surface, the FSC also retrieves snow grain size. Among other uses, rapid changes in snow grain size in the temporal domain can be used to infer the presence of clouds in ABI imagery.

In terms of the F&PS, the FSC is not directly responsible for other products. However, several AWG teams have identified a dependency upon *a priori* knowledge of snow cover. Additionally, since the ABI Cloud Mask (ACM) is dependent upon an ABI derived snow mask, any AWG product dependent upon the ACM may also be indirectly dependent upon the FSC.

The specific products generated by the FSC include:

- Fraction of pixel covered by snow (0.0 – 1.0) (primary),
- Fraction of pixel covered by non-snow surface and its land surface type (e.g., vegetation, bare soil, rock, etc. 0.0 – 1.0) (intermediate),
- Snow grain size (Sphere radii 10 to 1,100 μm) (intermediate),
- Binary (snow/not snow) coverage (intermediate)
- Quality values (intermediate)
- Quality flags (intermediate)
- RMSE retrieval confidence (intermediate).

Updates to the F&PS requirements for fractional snow state that the measurement accuracy for the FSC is 0.15 fractional and the measurement precision for FSC is 0.30 fractional (Table 1).

Table 1. Snow Cover product requirements from F&PS.

Region	Horizontal Resolution	Refresh Rate	Product Range	Product Accuracy	Product Precision	Product Type	Product Sub-type
Hemispheric	2 km	60 min	Fraction 0 – 1	0.15	0.30	Land	Land
Coterminous U.S.	2 km	60 min	Fraction 0 – 1	0.15	0.30	Land	Land
Mesoscale	2 km	60 min	Fraction 0 – 1	0.15	0.30	Land	Land

2.2 Instrument Characteristics

FSC will be produced for each noncloud pixel observed by the ABI. The GOESRSCAG model is dependent upon the spectral surface characteristics of the snow pack, as a function of snow grain size, in the visible and near-visible portion of the energy spectrum. Table 2 summarizes the ABI channel subset used by GOESRSCAG. Unless the design specification of the GOES-R ABI instrument changes, the final delivery channel subset used by the FSC has not changed as the algorithm is developed and validated.

It should be noted that since the FSC is based on the spectral signature of snow at the surface, the algorithm is designed to perform using surface reflectance corrected data generated by a rigorous Radiative Transfer Model (RTM). As of this writing the common spectral surface reflectance product is under development originally as a GOES-R option 2 product (Appendix 4, GOES-R Surface Reflectance Algorithm Theoretical Basis Document) with contributions primarily from the radiation/albedo team but also the cryosphere team. It is hoped that the development of an RTM for the GOES-R ground segment will benefit more than just the FSC algorithm's results.

Table 2. GOES-R ABI bands required in algorithm.

ABI Channel	Wavelength (μm)	Upper Limit of Dynamic Range	NEDT/SNR	Used in FSC?
1	0.45 – 0.49	625 W m^{-2} sr^{-1} μm^{-1}	300:1	✓
2	0.59 – 0.69	515 W m^{-2} sr^{-1} μm^{-1}	300:1	✓
3	0.8455 – 0.8845	305 W m^{-2} sr^{-1} μm^{-1}	300:1	✓
4	1.3075 – 1.3855	114 W m^{-2} sr^{-1} μm^{-1}	300:1	
5	1.58 – 1.64	77 W m^{-2} sr^{-1} μm^{-1}	300:1	✓
6	2.225 – 2.275	24 W m^{-2} sr^{-1} μm^{-1}	300:1	✓
7	3.8 – 4.0	400 K	0.1 K	*
13	10.1 – 10.6	330 K	0.1 K	*

* Use of thermal bands use is a planned enhancement

Secondly, since snow is a highly reflective material in the visible portion of the electromagnetic spectrum, the FSC may be sensitive to detector saturation and damping. Thirdly, since the FSC relies on spectral mixture analysis, imagery artifacts and instrument noise will negatively impact its performance.

The geometric fidelity of the GOES-R ABI may have an impact on the GOESRSCAG performance. If image-to-image pixel registration can be counted on (either mechanically or via software image navigation) the efficiency of the FSC's implementation can be enhanced by "buffering" certain GOESRSCAG calculations between temporally sequential data sets. This in particular will serve to enhance cloud/snow discrimination.

Finally, since FSC is an Earth surface feature, it is important to consider horizontal displacement distortions in the GOES-R ABI image data. While the end user may remove these distortions, it is preferable that parallax corrections be applied in the ground segment prior to product distribution. Correction of this distortion prior to the application of the FSC will nontrivially improve the algorithm's performance.

3. ALGORITHM DESCRIPTION

Complete description of the FSC at the current level of maturity (which will improve with each revision).

3.1 Algorithm Overview

The FSC retrievals serve a critical role in the GOES-R ABI processing system. It is a fundamental physical property but also serves to determine which pixels can be used for atmosphere and land cover applications (ACM, NDVI etc). The FSC is based on spectral mixing analysis. The implementation of the FSC, GOESRSCAG, has lineage directly from:

- HYPSCAG (Hyperion-based fractional snow cover and grain size) (Painter, 2002)
- MEMSCAG (AVIRIS-based fractional snow cover and grain size) (Painter et al., 2003)
- MODSCAG (MODIS-based fractional snow cover and grain size) (Painter et al., 2009)
- TMSCAG (Thematic Mapper-based fractional snow cover and grain size) (Painter et al., 2010b).

GOESRSCAG uses

- calculated (not directly measured) surface reflectance values
- calculated (not directly measured) cloud mask values.

Briefly stated, spectral mixture algorithms extract from the spectrum measured for a single pixel the proportions of individual spectra of the constituent materials (endmembers) observed by the instrument. The measured spectrum is proportionally decomposed into individual spectra by straightforward matrix inversion between the instrument observed spectrum and a library of *a priori* known, pure spectra.

GOESRSCAG, the implementation of the FSC for the GOES-R ABI, can be characterized as being a *Multiple Endmember Spectral Mixture Analysis* (Roberts et al., 1998a) wherein the

- Number of endmembers may vary pixel-by-pixel
- Endmembers themselves may vary pixel-by-pixel
- Snow endmembers are generated with radiative transfer model DISORT

which retrieves the following products

- Fraction of each pixel covered by snow (0.0 – 1.0) (primary)
- Fraction of each pixel covered by non-snow surfaces (e.g., vegetation, bare soil, etc.) (intermediate)
- Effective snow grain size of the per-pixel snow cover fraction (intermediate)
- Binary (snow/not snow) coverage (intermediate)
- Per-pixel RMSE retrieval confidence (intermediate)

In general, the fractional snow cover products can be used for clear conditions (snow mapping), cloudy conditions (cloud mapping), and as a pre-processing step for land cover pixel product generation for use by other applications.

3.2 Processing Outline

The processing outline of the core elements of the FSC, as expressed in the GOESRSCAG code set, is presented in Figure 1. The code set is designed to run on entire GOES-R ABI image sets. For processing efficiency, GOESRSCAG is implemented as a multi-threaded application with the number of threads being configurable.

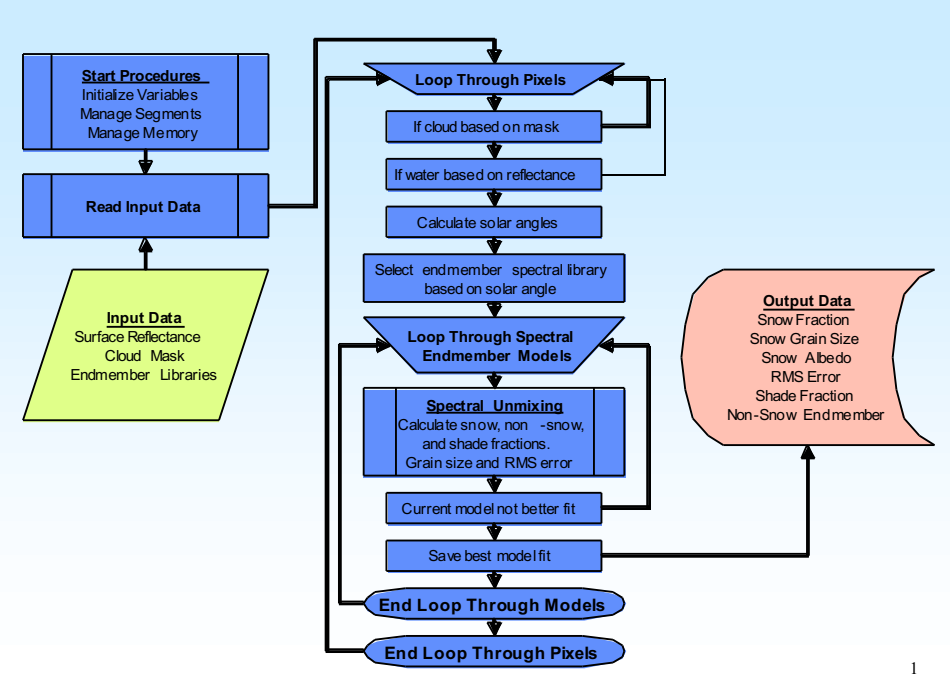


Figure 1. Basic flowchart of core snow cover elements in GOESSCAG implementation.

Two items should be noted. First, under certain circumstances GOESRSCAG is capable of making it own determination of cloud cover. GOESRSCAG cloud cover could augment the AWG Cloud Team ACM product.

Second, GOESRSCAG processing efficiency is enhanced by buffering certain calculations (e.g. previous FSC retrieval, pixel specific nonsnow endmembers) between time-sequential GOES-R ABI image sets. This aspect of the GOESRSCAG processing stream is not represented in Figure 1.

This section describes the input needed to process the FSC. The FSC is derived for each ABI pixel, and requires knowledge of clouds and surface reflectance. In its current implementation, we run the GOESRSCAG on each pixel.

Code will be described in much greater detail in Appendix 3 (Software Development Documentation) with important code sections detailed in subsequent appendices.

3.3 Algorithm Input

3.3.1 Primary Sensor Data

The list below contains the primary sensor data used by GOESRSCAG. By primary sensor data, we mean information that is derived solely from the ABI observations and geolocation information.

- GOES-R ABI surface reflectance bands
- Quality flags for surface reflectance bands
- Quality flags for general pixel utility
- View zenith θ_v and azimuth ϕ_v angles for each GOES-R pixel (fixed by satellite position – each is predictable)
- Solar zenith θ_0 and azimuth ϕ_0 angles for each GOES-R pixel (variable – each is predictable)
- First estimate of clouds generated locally or by the GOES-R Cloud Team. A snow-specific final cloud mask will be determined by the GOESRSCAG model using the grain size retrievals coupled with persistence metrics for changes in grain size.

3.3.2 Ancillary Data

The following data lists and briefly describes the ancillary data required to run GOESRSCAG. By ancillary data, we mean data that requires information not included in the ABI observations or geolocation data.

- Interpolated spectral libraries calculated ahead of time in five-degree increments of angles of solar zenith, view zenith, and relative azimuth (solar azimuth – view azimuth). These libraries are used to de-mix the spectral signature from a pixel. The various calculated spectra are used alone or in pairs to determine the best fit to the spectral response in the pixel.
- Model types are used to direct the algorithm on how to attempt to unmix a pixel's spectral signature. Currently, there are spectral libraries for many variations of snow, vegetation, rock, and ice.
- Dynamically-updated (by GOESRSCAG) non-snow endmember per pixel. The most-prominent non-snow endmember for each pixel is stored in a file, and on subsequent runs, the nonsnow endmember which corresponds to this ground covering is used to calculate the snow fraction. This field is recalculated for pixels which are close to solar noon (for best solar illumination). The non-snow endmember field is a design choice that reduces processing time. The initial search through endmember space must be comprehensive – e.g. we need to identify that it is vegetation mixed with the snow outside of your window. However, that tree is unlikely to move anytime soon and therefore a comprehensive search again through endmember space would be equivalent in effect to saying on a 15-30 minute basis, "is there still a tree outside my window?" However, trees do senesce, blow over, or find themselves subject to the hazards of the chainsaw. So, a period comprehensive search through the endmember space will

provide those necessary nudges to the non-snow endmember field. Remember, despite what many of our biologically-oriented remote sensing will tell you, snow exhibits the greatest range of spectral signatures of any surface on the planet and a range far greater than that of vegetation, even through senescence or soils through wetting and drying.

- Land-Water Mask. GOESRSCAG will skip non-land pixels identified in a common product land-water mask (Appendix 1, Common Ancillary Data Sets)
- The ABI 2-byte (4-level) cloud mask is used as input. The follow table can be used to map the Cloud mask (Appendix 1, Common Ancillary Data Sets):

Cloud condition:	FSCA response:
Clear	Model pixel
Probably Clear	Model pixel
Probably Cloudy	Model pixel, set PQI=PQI+200
Cloudy	Skip pixel

3.4 Theoretical Description

GOESRSCAG spectral mixture analysis derives from heritage algorithms that work on AVIRIS, AVHRR, and MODIS, retrieving subpixel fractional snow cover and grain size estimates via multiple endmember spectral mixture analysis (Appendix 2, Description of Modified Gram-Schmidt Orthogonalization). This physically based retrieval model is well-established and proven with MODIS (Painter et al., 2009) and AVIRIS (Painter et al., 2003) data. As is the case with any algorithm that uses optical data such as the reflectance bands of ABI, GOESRSCAG is unable to make snow retrievals under cloudy and heavily forested conditions. Implemented with MODIS surface reflectance data, the model has fractional snow cover uncertainty of < 0.05 and implemented with AVIRIS data has fractional snow cover uncertainty of < 0.04 (Painter et al., 2003; Painter et al., 2009).

3.4.1 Physics of the Problem

The current MODIS snow cover product, MOD10, is a "binary" map, whereby each pixel is classified as either "snow" or "not snow" (Hall, 2002). The algorithm's heritage traces back to retrieval of snow-covered area and qualitative grain size from the Landsat Thematic Mapper using normalized band differences (Dozier, 1989).

In contrast to the binary product, the GOESRSCAG model estimates the fraction of each pixel that is covered by snow, along with the grain size of that snow, using spectral mixture analysis and a radiative transfer model. Their simultaneous solution is necessary because the spectral reflectance of snow is sensitive to grain size (Warren, 1982) and the spectrum of the mixed pixel is sensitive to the spectral reflectance of the snow fraction (Painter et al., 1998). Therefore, we allow the snow's spectral reflectance to vary pixel-by-pixel and thereby address the spatial heterogeneity that characterizes snow cover in rough terrain (Painter et al., 2003; Painter et al., 2009).

3.4.1.1 Snow Endmembers

In spectral mixture analysis, an endmember is the spectral reflectance of a pure surface cover. GOESRSCAG uses a snow spectral library generated with model calculations of snow reflectance for monodispersions of spheres of radii 10 to 1,100 µm and solar zenith angles ranging from 0 degrees to 75 degrees of arc to accommodate changes in solar zenith angles during GOES-R ABI acquisitions. We calculate their single-scattering properties over each ABI band with Mie theory (Mie, 1908; Nussenzveig and Wiscombe, 1980; Wiscombe, 1980) and the hemispherical-directional reflectance factor R_λ (Schaepman-Strub et al., 2006) with a discrete-ordinates RTM (DISORT, Stamnes et al., 1988).

3.4.1.2 Rock, Soil, Vegetation, and Lake Ice Endmembers

The spectral library of vegetation, rock, soil, and lake ice comes from hyperspectral reflectance measurements made in the field and laboratory with an Analytical Spectral Devices field spectroradiometer (http://www.asdi.com). These spectra were convolved from 1 nm spectral resolution to the ABI bandpasses. **Error! Reference source not found.** Figure 2 shows snow (blue) and vegetation (red) endmembers contained in a spectral library with the GOES-R ABI and MODIS bandpasses indicated.

3.4.2 Mathematical Description

Linear spectral mixture analysis is based on the assumption that the radiance or reflectance measured at the sensor is a linear combination of radiances reflected from individual surfaces. The technique has been used to infer the fractional cover of vegetation cover (Roberts et al., 1998b; Okin, 2007), soils and rock cover (Asner and Heidebrecht, 2002; Ballantine et al., 2005), urban landscapes (Powell et al., 2007), and snow cover (Nolin et al., 1993; Rosenthal and

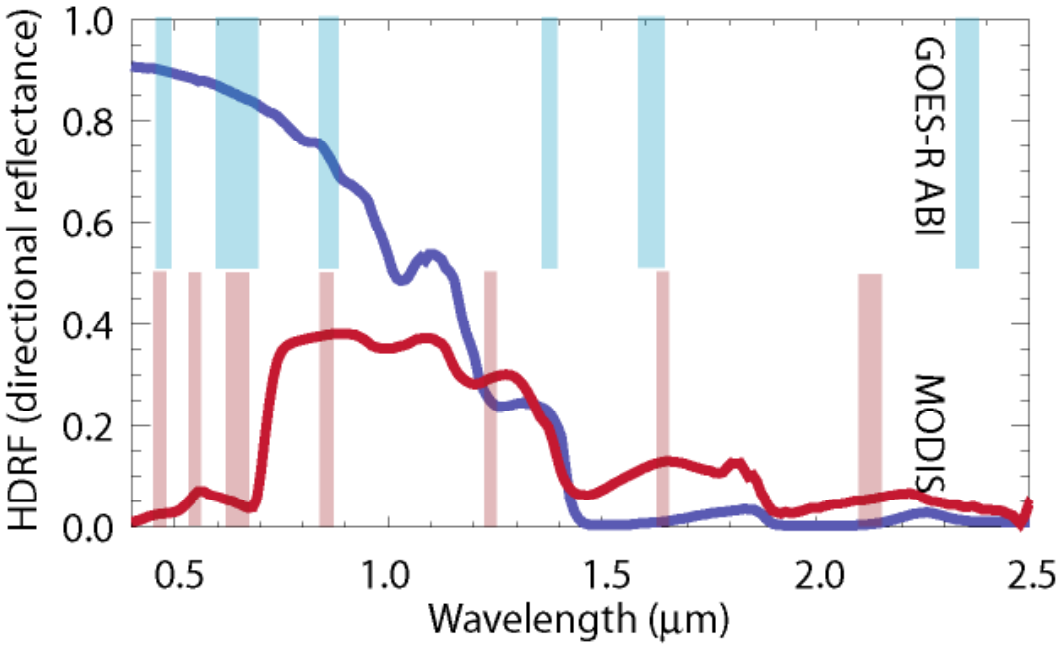


Figure 2. Snow (blue spectrum) and vegetation (red spectrum) endmembers with GOES-R ABI (blue bars) and MODIS (red bars) implementations.

Dozier, 1996; Painter et al., 1998; Painter et al., 2003; Painter et al., 2009). By contrast, Salomonson and Appel (2004; 2006) used a regression approach with the Normalized Difference Snow Index to infer fractional snow cover but with FSC uncertainties greater than 0.30 due to the tremendous scatter in the relationship between coarsened Thematic Mapper data and the NDSI (Painter et al., 2009a).

The linear assumption for spectral mixture analysis is appropriate for spatial scenarios such as snow and rock cover above timberline where the surface is near planar. Nonlinear analysis, which accounts for multiple scattering between surfaces, is necessary when the surface has a structure, such as vegetation that reflects and transmits radiation to the snow or soil substrate and other vegetation (Roberts et al., 1993). However, nonlinear mixtures can be linearized through the use of canopy-level endmembers. The vegetation endmembers in the spectral library are canopy-level measurements.

Spectral mixture analysis is based on a set of simultaneous linear equations that make up the components of the pixel-averaged ABI surface reflectance:

$$R_{S,\lambda} = \sum_{i=1}^{N} F_i R_{\lambda,i} + \varepsilon_\lambda \qquad (1)$$

where F_i is the fraction of endmember i, $R_{\lambda,i}$ is the hemispherical-directional reflectance factor of endmember i at wavelength λ, N is the number of spectral endmembers, and ε_λ is the residual error at λ for the fit of the N endmembers. The least squares fit arriving at F_i can be solved by several standard methods. The residual error is a rearrangement of the linear mixture model:

$$\varepsilon_\lambda = R_{S,\lambda} - \sum_{i=1}^{N} F_i R_{\lambda,i} \qquad (2)$$

The root mean squared error provides a spectrum-wide measure of fit for a mixture model:

$$RMSE = \left(\frac{1}{M} \sum_{\lambda=1}^{M} \varepsilon_\lambda^2 \right)^{1/2} \qquad (3)$$

where M is the number of imaging spectrometer bands used. The *RMSE* is a useful fundamental metric for optimizing selection of model results in the multiple endmember spectral mixture analysis (Dennison and Roberts, 2003).

The estimate of subpixel snow-covered area comes from the shade-normalized snow fraction f_S:

$$f_S = \frac{F_S}{\sum_{p \in S,v,r} F_p} = \frac{F_S}{1 - F_{shade}} \qquad (4)$$

where F_s is the snow spectral fraction, F_p are the physical spectral fractions (non-shade), and F_{shade} is the spectral fraction of photometric shade (Gillespie et al., 1990). Normalizing by the additive complement of the shade fraction accounts for topographic effects on irradiance. The estimates of subpixel vegetation cover, rock cover, and other surface cover are determined with equation (4) as well. The photometric shade is used for FSCA retrieval. Note that endmember_signatures has been extracted from the spectral libraries according to the ordering in

the model files and is not in the order of the spectral library. So, for a particular mixture model (say, snow of 100 μm and lodgepole pine 1), the endmember_signatures matrix is incremented as snow, vegetation, shade. Hence, the ordering.
The shade normalization that occurs in your first equation simply handles the fact that the mixture equation is unconstrained and shade is not treated in the mixture equation explicitly. Therefore, the physical fractions (snow, vegetation, soil) can have a sum that is not 1.0 and shade is solved for as the additive complement. The shade normalization however takes the spectral fractions of the physical constituents to their spatial fractions. So, for example, suppose we have snow spectral fraction of 0.5 and vegetation spectral fraction of 0.25 in a 3 endmember mixture. Then the fSCA = 0.5/(1-0.25) = 0.67.

3.4.2.1 GOESRSCAG Model

GOESRSCAG analyzes individual linear spectral mixtures for each permutation of two or more endmembers of the spectral library, in which no more than one endmember from a surface cover class is present (i.e., at most one snow endmember). For example, a potential model would consist of snow endmember of grain radius, coniferous forest, and photometric shade. A model is considered valid if: (a) spectral fractions are in the range [-0.01, 1.01], (b) overall RMSE is less than 2.5%, and (c) no three residuals exceed 2.5%. Mathematically, the greater the number of endmembers, the more trivial the solution (i.e. numerical nudging comes from extra vectors in providing solution within a given vector space). Therefore, any model that meets the modeling constraints in terms of (a) endmember fractions, (b) rms error, (c) residuals threshold, and (d) number of consecutive bands that can exceed that threshold, it is then preferable to select that with the lower number of endmembers and of that remaining set, select that with the lowest rms error.

(num of EMs) (tight|loose, 0|1) (frac min/zero/max pairs, 1pr / EM) (rms thresh) (rthresh) (rcount)
2, 1, -0.01 0.10 1.01, -0.01 0.10 1.01, 30.0, 30.0, 3
2, 0, -0.01 0.10 1.01, -0.01 0.10 1.01, 15.0, 15.0, 3
3, 1, -1.01 0.10 2.01, -1.01 0.10 2.01, -1.01 0.10 2.01, 50.0, 50.0, 3
3, 0, -0.01 0.10 1.01, -0.01 0.10 1.01, -0.01 0.10 1.01, 15.0, 15.0, 3

For each *n*-endmember suite of models that meet the constraints for a pixel, GOESRSCAG selects the snow fraction and grain size values associated with the smallest error and the tighter constraints. GOESRSCAG then attributes to the pixel the snow fraction and snow grain size of the valid model that has the fewest endmembers because a solution with more endmembers is mathematically trivial relative to that with fewer. The data flow of GOESRSCAG is exhibited in Figure 1.

GOESRSCAG incorporates the following assumptions: (a) the variability in the hemispherical-directional reflectance factor for the solar geometry and atmospheric conditions at the time of each GOES-R ABI acquisition is negligible, i.e., $R_\lambda(\theta_0,\phi_0,0,0) \approx R_\lambda(\theta_0,\phi_0,\theta_r,\phi_r)$ within the range of angles $[\theta_r,\phi_r]$ observed from GOES-R ABI; (b) the effects of impurities and the effects of thin snow on snow spectral reflectance are not separable and these effects do not impact retrievals of snow area and grain size; (c) linear spectral mixture analysis is valid for multispectral scenes of alpine terrain; and (d) liquid water in the snow does not affect the retrievals of snow-covered area and grain size. Painter et al. (2003) and Painter and Dozier

(2004) specifically confirmed the validity of these assumptions for spectral mixture analysis of FSC.

3.4.3 Algorithm Output

There are three types of final output that are produced by the algorithm: products (snow fraction), quality information (quality values, quality flags), diagnostic/intermediate information (non-snow fraction, snow grain size, binary snow, vegetation fraction, soil fraction), and metadata (RMSE). Table 3 gives information about each of these outputs.

Table 3. Snow cover (GOESRSCAG) algorithm outputs

Output	Type	Description
Snow Cover Fraction	Product Data	Fraction of pixel covered by snow endmember (0.0 to 1.0)
Product Data Quality Flags	Product Data	
Product Metadata	Product Data	
Quality Value/Product Quality Information (Diagnostic)	Product Data	
Binary Snow Mask	Intermediate Data	Pixel is covered/not covered by snow (0 or 1)
Dominant Non-Snow Endmember Type	Intermediate Data	
Snow Grain Size	Diagnostic Data	Sphere radii ranging from 10 to 1,100 μm
RMSE of Retrieval	Diagnostic Data	Retrieval confidence (0.0 to 1.0)
Vegetation Fraction	Intermediate Data	Fraction of pixel covered by vegetation (0.0 to 1.0)
Rock/Soil Fraction	Intermediate Data	Fraction of pixel by rock or bare soil (0.0 to 1.0)
Other Fraction	Intermediate Data	

Table 3. Product output and data type for each output.

The algorithm does produce quality flags. Table 4 lists those quality flags.

QA bit	Meaning if bit is set
0	No-data value in band data
1	Missing data in band data
2	Modeled cloudy
3	Salt water
4	Solar zenith angle out of acceptable range
5	Sensor zenith angle out of acceptable range
6	Bad metadata or ancillary data
7	Other reason

Table 4. Interpretation of QA bits in quality bits field.

Table 5 contains a list of conditions which would trigger the various quality flags and set the Product Quality Information flags.

Condition:	DQF set	PQI set to:	FSCA
Missing metadata	1 (bit 0)	0.00	Skip pixel
Any band's reflectance value is missing	1 (bit 0)	0.00	Skip pixel
Any band's reflectance value is bad, saturated,	2 (bit 1)	5.00	Skip pixel
Absolute value of latitude > 90° or absolute	64 (bit 6)	6.00	Skip pixel
LZA is < 0° or > 90°	32 (bit 5)	7.00	Skip pixel
SZA is < 0°	16 (bit 4)	2.00	Skip pixel
SZA is > 90°	16 (bit 4)	3.00	Skip pixel
SZA is > 67.5° and <= 90°	16 (bit 4)	8.00	Proceed
Pixel is over water	8 (bit 3)	1.00	Skip pixel
Cloud covered based on ABI Cloud Mask	4 (bit 2)	Add 100.00	Proceed
LZA is > 67.5° and <= 90°	32 (bit 5)	Add 10000.00	Proceed
Clear	#	#	Pixel
Probably Clear	#	#	Pixel
Probably Cloudy	4 (bit 2)	Add 200.0	Pixel
Cloudy	4 (bit 2)	Add 100.0	Pixel

The primary output of the algorithm is (fractional) snow cover. Along with the header file that will accompany the data file, there will be additional metadata produced for the image. For the entire image, the algorithm will provide the RMS, maximum, and minimum values for those areas in an ASCII list. The following is an example of this metadata for the entire image ("Full Disk"):

Region name: Full Disk
Minimum fraction: <maximum value>
Maximum fraction: <minimum value>
Mean faction: <mean value>
Mean of RMS: <RMS value>
Standard deviation of RMS: <stddev value>
Number of QA flags: <value>
Flag 0 QA description: <string>
Flag 0 % of retrievals: <value>
Flag 1 QA description: <string>
Flag 1 % of retrievals: <value>
…
Flag 7 QA description: <string>
Flag 7 % of retrievals: <value>

Quality assurance (QA) values will be tracked for each pixel in both the entire image. These values (in the quality_bits file) will be composed of eight bits that can be interpreted as an integer. If all bits are "turned off" (i.e., the QA value is 0), a good snow-fraction retrieval was performed. This can also include a successful retrieval of no snow (0%). If a pixel is not modeled, or is modeled but the snow fraction value should be used with caution, its QA value will not be 0. Table 4 interprets the use of the quality bits. Bit 0 is the least-significant bit of the QA value; bit 7 is its most significant bit.

The format of these data will be of the form "variable: value", where "variable" will be the name of the datum (such as "whole image RMS" or "region 1 maximum") and "value" will be the value of that particular item.

The GOESRSCAG algorithm has a 60 minute refresh, therefore it should be run once an hour.

The product quality information associated with each FCSA retrieval is defined as follows:

0.00: if the no data value was encountered in any of the inputs, or

1.00: if the pixel falls on water, or

2.00: if the solar zenith angle is too small, or

3.00: if the solar zenith angle is too large, or

4.00: if the pixel could not be modeled, or

5.00: if the pixel had bad source data, or

6.00: if the pixel has bad horizontal location (latitude or longitude) metadata, or

7.00: if the pixel has an unreasonable sensor angle value, or

8.00: if the pixel has an unreasonable solar zenith angle, or

10.00: if the pixel could be modeled but is snow free, or

2f.ff: if the pixel could be modeled and has snow (where fife is the snow fraction), or

3f.ff: if the pixel could be modeled and has shaded snow (zero grain size) (should be physically impossible but is mathematically feasible) (where f.ff is the snow fraction), or

4f.ff: if the pixel could be modeled and has cloud by grain size (where f.ff is the cloud fraction).

The flags may be modified by the FSCA for cloud masking as follows:

Add 100.00 if cloud masked, or

Add 200.00 if probably cloud masked, or

The flags may be modified by the FSCA for the following second-most prominent non-snow endmember in a modeled pixel:

Add 1,000.00 if the most prominent non-snow endmember is vegetation, or

Add 2,000.00 if the most prominent non-snow endmember is vegetation, or

Add 8,000.00 if the most prominent non-snow endmember is other, or

Add 9,000.00 if the most prominent non-snow endmember is photometric shade. *S*hade normalization is done before the most prominent nonsnow **ground** cover is determined, and this is done by comparing the different types of **ground** cover – vegetation, rock, lake ice – against each other and determining which fraction is largest. The scenario of photometric shade being the most prominent non-snow endmember occurs only when the pixel is 100% snow.

The flags may be modified by the FSCA for the following condition:

Add 10,000 if the pixel was modeled but beyond the sensor angle threshold

4. TEST DATA SETS AND OUTPUTS

GOESRSCAG requires as input the spectrum measured by the Advanced Baseline Imager. Because ABI is a quantum step forward in geostationary sampling, no data exist from GOES or SEVERI to serve as proxy data. Therefore, in order to perform validation in the Pre-launch Phase, we must use proxy data from the current MODIS instruments on Terra and Aqua while still available, and, once no longer available, data from the Visible Infrared Imaging Radiometer Suite (VIIRS) in the NPOESS Preparatory Project and the NPOESS. In the Post-launch Phase,

we will perform validation of the algorithm directly from the ABI data and with the VIIRS data in order to assess how accurately proxy data had represented the ABI data in the Pre-launch Phase.

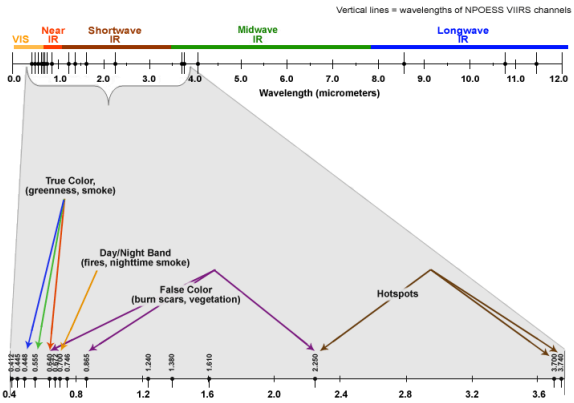


Figure 3. VIIRS bandpasses. Note these include some bands that are not included in the surface reflectance products.

4.1 Input Data Sets

Proxy data will be created with MODIS data for as long as the Terra and Aqua MODIS instruments are functional. As of Spring 2009, both have exceeded their design lives of six years (Terra in 2006, Aqua in 2008), so it is likely that no MODIS acquisitions will be available by the 2015 launch of GOES-R. However, radiance and surface reflectance data from the NPOESS Preparatory Project (NPP) Visible/Infrared Imager Radiometer Suite (VIIRS) will be available from the NOAA Comprehensive Large Array-data Stewardship System (CLASS) and NASA EOSDIS (Figure 3). Launch as of this writing is scheduled for 2010. Data from the NPOESS VIIRS should be available in 2013 according to present launch schedule but this date may well slip if NPP is an indication of readiness for NPOESS. While input data from MODIS data has been thoroughly tested regarding its use in the algorithm, it is expected that the other aforementioned data sources will be compatible with the algorithm or could be made so with minimal adjustments to the data.

Proxy and simulated ABI data will be made available by the GOES-R proxy data team. During Pre- and Post-launch periods, the NOHRSC will be running MODSCAG/VIIRSSCAG operationally. Therefore, we can ship the either the raw data or the FSC retrievals to the AWG proxy data team for most teams to use as proxy FSC for their algorithms. The FSC team requires proxy data that mimic the ABI spectrum of bands 1 through 3 and 5 through 6. MODIS bands 1, 3, 4, 6, and 7 and the VIIRS bands M3, M5, M7, M10, and M11 are analogues for the required bands in the ABI spectrum (Table 4).

Already we have begun collections of MODIS calibrated radiance data (Level 2) at the National Operational Hydrologic Remote Sensing Center (NOHRSC). We download MOD02IKM, MOD02HKM, MOD02QKM and MOD03 level 1b data and convert them to surface reflectance with the MOD09_SPA program that is made available by the NASA Direct Readout Laboratory (http://directreadout.sci.gsfc.nasa.gov - version V5.3.18). The MOD09_SPA code converts the radiance data to surface reflectance with a sparser ancillary dataset that characterizes the atmosphere so that surface reflectance is available with latency

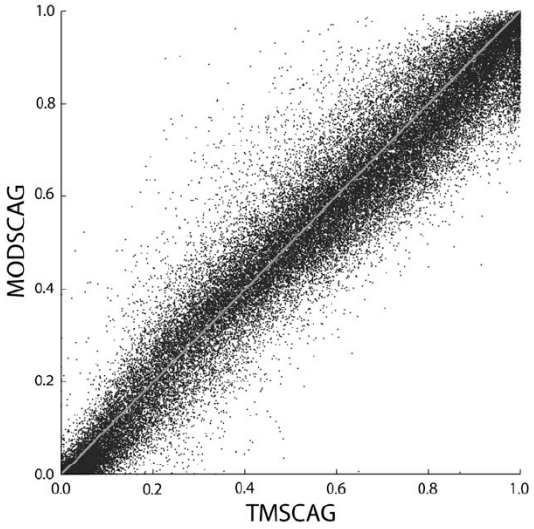


Figure 4. Aggregated comparison of MODSCAG and TMSCAG.

of less than 1 day rather than the approximately four-day latency of delivery of the MODIS MOD09GA Surface Reflectance Product. MOD09_SPA uses the Global Data Assimilation Group (GDAS) numerical weather prediction model. NOHRSC has a bent-pipe feed of MODIS radiance data (MOD02x, MOD03x) from NASA EOSDIS and GDAS data from the Goddard Space Flight Center so that we can access these data at sub-day latency. The greatest lag comes in the delivery of GDAS data with latencies as great as six hours.

Table 4. FSC GOES-R ABI proxy data for pre-launch (MODIS, VIIRS) and post-launch

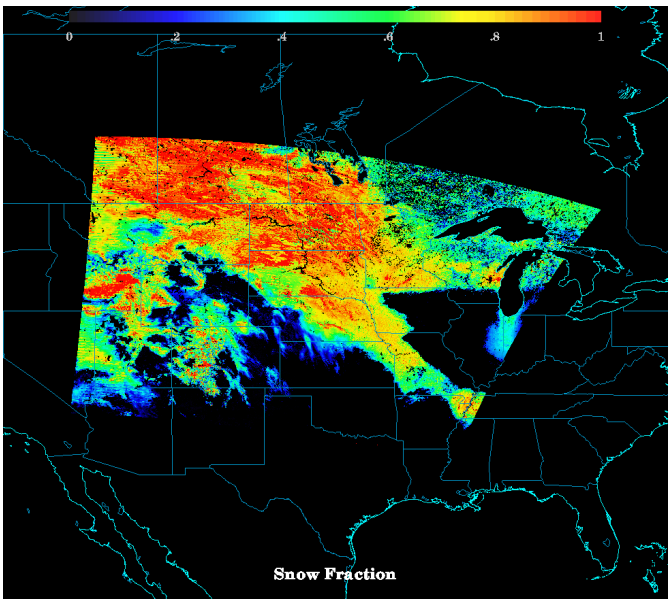
GOES-R ABI Channel Number	GOES-R ABI Wavelength (μm)	Used in FSC	MODIS proxy Channel	VIIRS Proxy Channel
1	0.47	✓	1	M3
2	0.64	✓	3	M5
3	0.8655	✓	4	M7
5	1.61	✓	6	M10
6	2.25	✓	7	M11
7	3.9	*	21	M12
13	10.35	*	31	M15

* Use of thermal bands use is a planned enhancement

We will use the MODSCAG model to retrieve FSC for validation from the seven band MODIS surface reflectance data (Painter et al., 2009). In parallel, the MODIS data will be reduced in band space to that of ABI as shown above. Given that MODSCAG has an FSC accuracy of -0.005 and precision of 0.049 as shown in **Error! Reference source not found.** and in Error! Reference source not found. (Painter et al., 2009), 7-band FSC retrievals from MODIS will represent a validation set that addresses the model uncertainty that is related to the changes in band space at native resolution of 500 m and then coarsened to 2 km.

Upon the availability of NPP and NPOESS VIIRS reflectance data, we will use ABI proxy data from NPP VIIRS and NPOESS VIIRS data. As is done with MODIS data, we will initially run the new model (VIIRSSCAG) with the 11 VIIRS reflectance bands M1-M11, which span 0.412 to 2.25 μm wavelength, at 742 m spatial resolution, for validation. Subsequently, we will compare the 5-band proxy ABI retrievals with the full VIIRSSCAG retrievals at the native resolution of 742 m and then coarsened to 2 km. Note that Investigator Painter is discussing with the Northrup-Grumman team in charge of algorithms for VIIRS the adoption of VIIRSSCAG as the standard snow cover product for NPOESS VIIRS. NPP VIIRS already has adopted the more simplistic binary snow cover model based on poor implementation of the original ATBD for snow cover for VIIRS. Table 4 describes the GOES-R channels used in the GOESRSCAG algorithm and their corresponding bands in proxy data from other satellites.

Validation of the 5-band proxy ABI data against the MODSCAG 7-band results has been performed with several years' data for the coterminous US (2008 – 2010). Merging of all of these scenes including 38 full CONUS retrievals from weekly reflectance composites in 2010 from MODIS showed that FSC (GOESRSCAG) has accuracy well within specification. These results are described below.



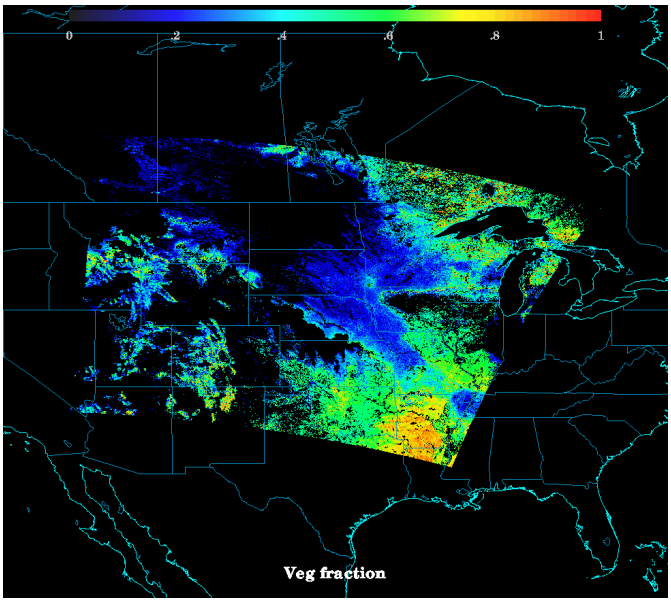


Figure 5. Simulated GOES-R ABI snow fraction (top) and green vegetation fraction (bottom) from GOESRSCAG processing of proxy ABI data from MODIS, 1 March 2009.

4.2 Output from Input Data Sets

The output data from the proxy ABI data from MODIS have the same general formats as those that will come from the operational ABI processing. These will be represented as continuous fractional snow cover (as well as fractional green vegetation, fractional soil/rock/senesced vegetation) as shown in **Error! Reference source not found.** for the north-central coterminous US into southern Canada. These output data match the content of the algorithm package delivery.

4.2.1 Accuracy and Precision Estimates

The mean difference between the five bandpass GOESRSCAG outputs and the seven bandpass MODSCAG outputs were calculated where both algorithms identified snow on the same pixel (Table 5). Calculation of difference for all regions that also include no snow was calculated and establishes a better estimate of the detection of snow than the quantification of snow covered area (Table 5).

4.2.2 Error Budget

Using the requirements shown in Table 1, the snow cover algorithm meets the F & PS 100% requirements.

The text has described the first two rows of Table 5 in describing the MODSCAG 7 band model. For the comparison of the 5-band ABI proxy version relative to MODSCAG, we now give the results. The overall accuracy (pixel-weighted, mean difference) shown in this phase of testing and validation was 0.023 (snow and snow-free) and 0.037 (snow only) across all scenes described above. Therefore, FSC (GOESRSCAG) meets the specification

Table 5. Validation results of FSC (GOESRSCAG) under various scenarios.

Validation Configuration	Accuracy (Spec)	Precision (Spec)
Fractional snow cover MODSCAG vs. Landsat (snow only)	-0.010 (0.15)	0.089 (0.30)
Fractional Snow Cover MODSCAG vs. Landsat (snow and snow-free)	-0.005 (0.15)	0.049 (0.30)
Fractional Snow Cover 5-band ABI proxy vs. 7-band MODSCAG (snow only)	0.037 (0.15)	0.119 (0.30)
Fractional Snow Cover 5-band vs. 7-band ABI proxy (snow and snow-free)	0.023 (0.15)	0.077 (0.30)

of 0.15 with a buffer of ~0.12. It is highly unlikely that any algorithm modifications would push the accuracy into that buffer. Given the accuracy result of the MODSCAG algorithm, the greatest composite accuracy value for FSC (GOESRSCAG) is 0.018 (snow and snow-free) and 0.027 (snow only), again well within specification of 0.15.

The overall precision for the 5-band ABI proxy data in this phase of testing and validation was 0.077 (snow and snow-free) and 0.119 (snow only) across all scenes described above (Table 5). Therefore, FSC (GOESRSCAG) meets the specification of 0.30 with a buffer of ~0.20. It is highly unlikely again that any algorithm refinement would increase precision values through that buffer. Given the precision result of the MODSCAG algorithm, the worst-case composite precision values for FSC (GOESRSCAG) lie < 0.20, again well within F&PS requirements of 0.30.

In addition, in Table 6 are the errors of omission and commission from the tests. Two hundred forty-three MODIS granules were processed with the SCAG algorithm. These granules covered an area bounded by parallels of latitude 24ºN and 60ºN and meridians of latitude from 65ºW to 126ºW, covering the coterminous U.S. and southern and central Canada. The time period used was from 2009 October 1 to 2010 June 30. This area is where all testing of the algorithm has taken place.

4.3 Numerical Computation Considerations

The FSC relies on two primary inputs:

- Surface reflectance values for GOES-R ABI channels 1, 2, 3, 5 and 6 (and eventually 7 and 13)
- An *a priori* calculated cloud mask

Table 6. Raw counts of errors in test data.

Statistic	Value
Number of pixels processed	668 million
Number of errors of commission	3.02 million (0.045%)
Number of errors of omission	2.92 million (0.045%)

Some would argue that snow, from a spectral analysis perspective, behaves like a low-altitude, large-particle cloud. Using this supposition,

it is possible to leverage the spectral mixture analysis for FSC and the rapid GOES-R image acquisition schedule to assist the AWG Cloud Team's ACM product. Temporal signatures of grain size can provide an additional capability for cloud masking as short terms changes in grain size indicate cloud presence. Since the FSC (which depends on a cloud mask) and the ACM (which depends on a snow mask) are interdependent, it seems reasonable to pursue synergies between the two algorithms.

FSC as expressed in GOESRSCAG relies on linear transformations to decompose a pixel's spectral signature into its constituent spectra. GOESRSCAG employs (or will employ) several strategies to reduce its computational load:

- Limit the number of possible endmembers to two (snow and a single non-snow endmember from a limited list of possibilities) plus shade. GOESRSCAG retrieves, for example, combinations of snow and vegetation or snow and bare ground. Since the primary aim is the determination of FSC (as opposed to the identity and proportion of the non-snow constituents), this limit minimizes the dimensions of the matrices. However, GOESRSCAG has the full capacity to map combinations of vegetation and soil, and as such, can act as a robust cross-validation of other vegetation retrievals. (Status: done)
- Limit the number of possible snow grain size spectra (Status: done)
- Limit the number of spectra within a possible non-snow endmember spectrum library (Status: done)
- Integrate (and then optimize) spectrum mixing analysis modeling with the final endmember sorting/selection logic execute a FSC retrieval with as few matrix operations as possible (Status: done)
- Buffer repetitive, intermediate calculations between time sequential GOES-R ABI images (e.g., the identity of the non-snow endmember for a given pixel). (Status: in progress)

4.4 Programming and Procedural Considerations

The GOESRSCAG FSC is a pixel-by-pixel algorithm (Appendix 3, Software Development Documentation) that will benefit by running on time-sequential images. The algorithm relies on matrix operations that impact program design and implementation. While this reliance impacts programming considerations, we addressed strategies for its mitigation above.

Additionally, we recognize that the GOES-R satellite will not be deployed for several years. With that in mind, rather than adopting a more traditional single-threaded architecture, we have implemented the GOESRSCAG FSC as a multi-threaded application more suitable to the multiprocessor computers anticipated for the near future.

4.5 Quality Assessment and Diagnostics

The following procedures are recommended for diagnosing the performance of the FSC.

- Monitor the percentage of specific endmembers in regional areas where these values should be nearly constant after snow cover reaches 100 percent during accumulation in the fall or snow cover reaches 0 percent during ablation in the spring.
- Assess persistence/consistency of fractional snow cover by pixel. There should be no rapid oscillations in FSC and grain size for a given pixel. FSC for a given pixel should vary smoothly in the temporal domain except immediately after cloud cover that has produced snowfall.
- Assess errors of confusion between cloud cover and snow cover.
- Assess fractional snow cover retrievals with high spatial resolution, polar-orbiting sensors such as the Landsat Thematic Mapper.
- Assess fractional snow cover retrievals with physically based, energy- and mass-balanced snow models.
- Periodically review the individual test results for artifacts or non-physical behaviors.
- Maintain close collaboration with other teams using the FSC in their product generation.
- Maintain a close collaboration with the cloud teams to resolve issues associated with snow/cloud discrimination.

4.6 Exception Handling

The GOESRSCAG FSC will include checking the validity of each required channel before executing its retrievals. The GOESRSCAG FSC also expects the Level 1b processing to flag any pixels with missing geolocation or viewing geometry information. The following additional pixel-by-pixel exceptions will be identified and flagged by the FSC in its output:

- Clouds identified by ACM and/or grain size
- Pixels below the solar zenith angle threshold
- Pixels that are saturated
- Pixels missing surface reflectance RTM correction
- Pixels too close to limb.

In these cases, appropriate flags will indicate that no FSC retrieval was made for that pixel.

4.7 Algorithm Validation

FSC is a quantitative, area representation of snow cover rather than a simple detection of presence somewhere in the area. Therefore, it is necessary to assess the spatial heterogeneity in the algorithm errors with validation data that have the capacity to reveal the distribution of snow cover at a scale finer than that of GOES-R and that can be coarsened to the distribution at the scale of GOES-R. The retrievals from GOESRSCAG will be assessed in terms of their fractional accuracy, fractional precision, and stability over space and time. We will also assess their binary accuracy for those users who may be interested in simple detection of snow presence. The validation of GOESRSCAG with the ABI proxy data from MODIS and VIIRS will address the accuracy and precision of the model. The validation of GOESRSCAG with the high spatial resolution data will facilitate understanding the subpixel drivers of uncertainty in FSC retrievals such as anisotropic distribution of vegetation and topographic variation that affects irradiance and snow grain size distributions.

In either of the pre-launch or post-launch periods, the primary validation datasets will be retrievals of snow cover from medium and high spatial resolution polar orbiting data, the scales of which are greater than that of GOES-R ABI (2 km). In the Pre-launch Phase, the medium resolution data will come from MODIS (500 m) and the high-resolution data from the Landsat-5 Thematic Mapper and Landsat-7 Enhanced Thematic Mapper Plus (ETM+) (29 m) (Figure 6Figure 7). In the Post-launch period, the medium resolution data will come from VIIRS (370 m) and the high-resolution data from Landsat Data Continuity Mission (LDCM) (29 m). Secondary validation will come in the form of binary validation with in situ measurements from cooperative observer networks and snow pillows and courses, numerical snow model state variables, as well as from the higher resolution satellite instruments.

The GOESRSCAG algorithm will be evaluated with two levels of data sets; MODIS and VIIRS proxy data at 500 m and 2 km spatial resolution and coincident high spatial resolution data from the Landsat Thematic Mapper (TM). The model will be assessed based on its FSC accuracy and precision. Accuracy and precision will be evaluated according to the F&PS (Table 1).

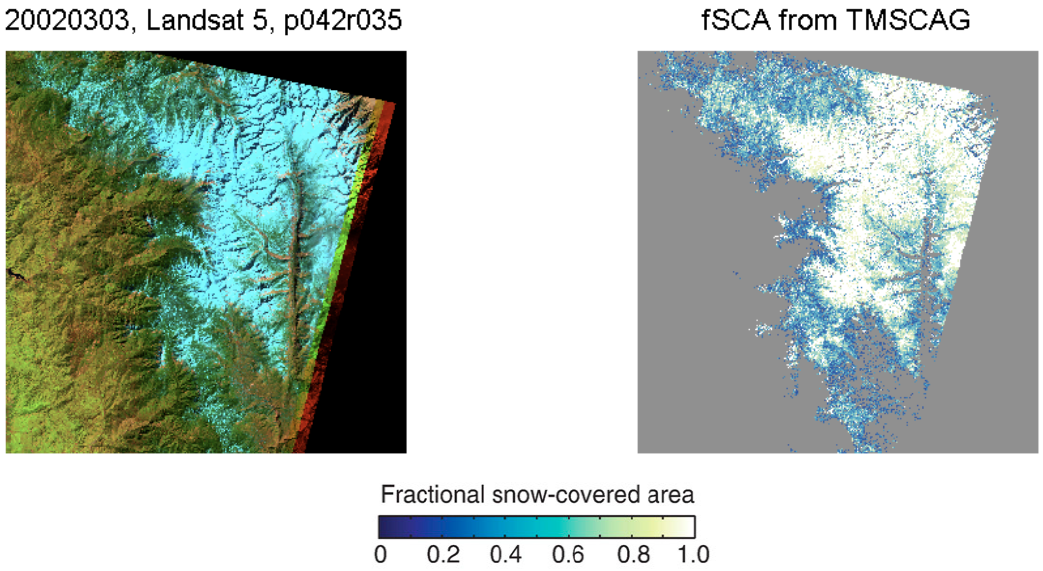


Figure 6 TMSCAG results for the southern Sierra Nevada (Sequoia and Kings Canyon National Parks), California. (left) Color composite of TM5 bands. (right) FSC for the same scene.

The sensitivities of the algorithm will be assessed with the high-resolution fractional snow cover data and ancillary data. The dependent variables will be per pixel FSC errors and the independent variables will be solar zenith angle, local zenith angle, sensor azimuth angle (relative to solar principal plane), uncertainty in surface reflectance retrieval, elevation, aspect, topographic variance, land cover, vegetation type, and snow grain size. Perfect success of the algorithm would be shown by insignificance in the relationships above with near zero errors in FSC. However, perfect success is highly unlikely given the chain of uncertainties from sensor radiance response through atmospheric characterization and surface reflectance retrieval to model uncertainties. While the regressed relationship between RMSE error and solar zenith angle is positive (consistent with our hypothesis), the errors are relatively insensitive with the R^2

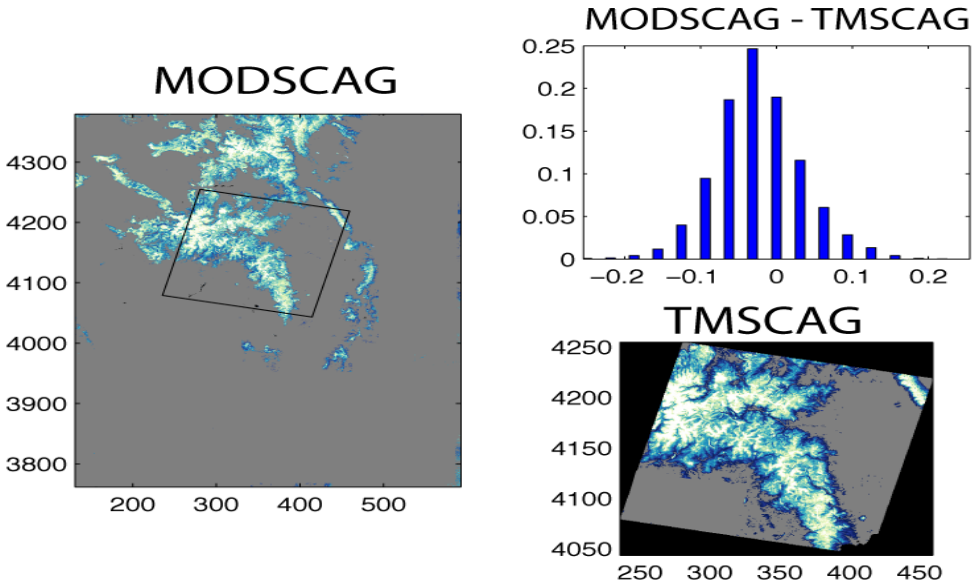


Figure 7. MODSCAG validation with TMSCAG. This scene shows the Upper Rio Grande basin of Colorado and New Mexico, on 17 April, 2001. The histogram presents the distribution of errors.

= 0.22. The reader at this point is most likely considering algorithm stability in time. This will be described in detail below and treated during the Post-launch Phase.

The full details of the algorithm validation are handled in the **Product Validation Plan Document: *Fractional Snow Cover***. Below we describe the pre-launch phase activities and post-launch phase activities.

4.7.1 Pre-launch Phase Activities

The NOHRSC will continuously ingest and correct MODIS (VIIRS when available) level 1B data for the CONUS, Canada, and Alaska in real time. Both 7-band (validation) and 5-band (ABI proxy) FSC retrievals will be produced and analyzed as described above. A time series of validation statistics, computed daily, will characterize the accuracy, precision, and spatiotemporal-stability of the FSC. In addition to these moderate resolution validation efforts, ancillary validation efforts will conducted (as described above) using high resolution Landsat data, the NOHRSC's numerical snow model, and the NOHRSC's comprehensive collective of ground-based snow observations. Modifications to the FSC will be made by Painter/Rost/Eicher as necessary during the pre-launch phase.

4.7.2 Post-launch Phase Activities

As described above, the post-launch allows us an opportunity to replace proxy ABI with real ABI data. Early post-launch activities will consist of the same activities conducted during the pre-launch phase with the replacement of proxy data with real data. Once validated (and adjusted as needed) the FSC will be considered deployed. A subset of the pre-launch validation activities will be continued by the NOHRSC in post-launch and beyond as part of its operational mission. These results can be fed forward to the GOES-R FSC deployment to serve as a real time cross validation dataset. It should be noted the FSC uses external spectral signature files to drive its numerical analysis. Refinements to the model's performance can be easily affected by replacing these files. Efforts at the NOHRSC to improve its MODIS/VIIRS implementation of the FSC can

be leveraged by GOES-R by this means. Critical to validation in the Post-launch Phase will also be the extensive stability testing at the daily scale and sub-day scale.

5. ASSUMPTIONS AND LIMITATIONS

The following sections describe the current limitations and assumptions in the current version of the GOESRSCAG FSC.

5.1 Performance

The following assumptions have been made in developing and estimating the performance of the FSC. The following list contains the current assumptions and proposed mitigation strategies:

- Background snow-free surface reflectances will be available based on our preprocessing.
- Horizontal displacement distortions removed by parallax correction. Otherwise relax constraints on solar angle-geometry calculations
- The processing systems allows for ingest of previous output for application of the temporal tests. Otherwise allow for reduced performance of GOESRSCAG and give up GOESRSCAG cloud analysis
- High-quality cloud maps are available. Otherwise calculate cloud mask within the FSC

Note that FSC retrievals can only be made where snow is visible to the sensor. Snow under trees cannot be retrieved once snow falls off of the canopy. This can be partially mitigated by keeping track of forested areas during the snow-free season and tracking canopy intercepted snow and carrying it through subsequent images.

The algorithm depends solely on data from GOES-R; there is currently no plan to incorporate data from degraded or missing source layers from other data sources. Degraded data – such as a "noisy" band – will cause degraded products; missing data, including ancillary or metadata, will cause the algorithm to not produce output but fail gracefully.

5.2 Assumed Sensor Performance

We assume the sensor will meet its current specifications. However, the GOESRSCAG will be dependent on the following instrumental characteristics:

- Unknown spectral shifts in some channels will cause biases in the clear-sky surface reflectance calculations that may impact the performance of GOESRSCAG.
- Errors in navigation from image to image will affect the performance of the temporal tests.
- Any saturation in the visible wavelength channels will effect performance of mixture analysis
- Loss of a band can degrade performance of GOESRSCAG. In particular, loss of ABI band 5 would lose leverage for grain size retrieval and, in turn, cause the loss of cloud-masking capability.
- Temporal analysis in FSC will be critically dependent on the amount of striping in the data.

5.3 Pre-Planned Product Improvements

The FSC performance must be optimized for the Land Application Team, Cloud Team and Hydrology Team algorithms. We therefore intend to allow for feedback and to incorporate any suggestions from them to improve the FSC. In particular, we feel that the interdependence between the FSC and the ACM requires special attention and coordination, perhaps through the AIT.

The FSC serves many other applications. Its development is therefore tied to the development and feedback from the other algorithms. At this point, it is therefore difficult to predict what the future modifications will be. However, the following list contains our current best guess of the future FSC modifications:

- Temporal signatures of snow cover and grain size for cloud mapping applications and quality control
- Unless addressed elsewhere (e.g., by the AIT), some of the performance mitigations addressed above.

Additionally, we will continue to cooperate with the AIT to pursue

- RTM-driven surface-reflectance correction
- Parallax correction for horizontal displacement distortions.

6. REFERENCES

Asner, G. P., and Heidebrecht, K. B. (2002), Spectral unmixing of vegetation, soil and dry carbon cover in arid regions: comparing multispectral and hyperspectral observations, *International Journal of Remote Sensing*, 23(19): 3939-3958, doi: 10.1080/01431160110115960.

Ballantine, J.-A. C., Okin, G. S., Prentiss, D. E., and Roberts, D. A. (2005), Mapping North African landforms using continental scale unmixing of MODIS imagery, *Remote Sensing of Environment*, 97(4): 470-483, doi: 10.1016/j.rse.2005.04.023.

Dennison, P. E., and Roberts, D. A. (2003), Endmember selection for multiple endmember spectral mixture analysis using endmember average RMSE, *Remote Sensing of Environment*, 87(2-3): 123-135, doi: 10.1016/S0034-4257(03)00135-4.

Dozier, J. (1989), Spectral signature of alpine snow cover from the Landsat Thematic Mapper, *Remote Sensing of Environment*, 28(1): 9-22, doi: 10.1016/0034-4257(89)90101-6.

Gillespie, A. R., Smith, M. O., Adams, J. B., Willis, S. C., Fischer III, A. F., and Sabol, D. E. (1990), Interpretation of residual images: spectral mixture analysis of AVIRIS images, Owens Valley, California, *Second Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Workshop,* Pasadena, CA, Jet Propulsion Laboratory, 90-54, pp. 243-270.

Hall, D. K., G. A. Riggs, V. V. Salomonson, N. E. DiGirolamo, and K. J. Bayr (2002), MODIS snow-cover products, *Remote Sensing of Environment*, 83: 181-194.

Mie, G. (1908), Beiträge zur Optik trüber Medien, Speziell Kolloidaler Metallösungen, *Annalen der Physik*, 25: 377-445.

Nolin, A. W., Dozier, J., and Mertes, L. A. K. (1993), Mapping alpine snow using a spectral mixture modeling technique, *Annals of Glaciology*, 17: 121-124.

Nussenzveig, H. M., and Wiscombe, W. J. (1980), Efficiency factors in Mie scattering, *Physical Review Letters*, 45(18): 1490-1494.

Okin, G. S. (2007), Relative spectral mixture analysis—A multitemporal index of total vegetation cover *Remote Sensing of Environment*, 106(4): 467-479, doi: 10.1016/j.rse.2006.09.018.

Painter, T. H., Roberts, D. A., Green, R. O., and Dozier, J. (1998), The effect of grain size on spectral mixture analysis of snow-covered area from AVIRIS data, *Remote Sensing of Environment*, 65(3): 320-332, doi: 10.1016/S0034-4257(98)00041-8.

Painter, T. H. (2002), *Cold Land Processes Field Experiment. Hyperion data*. National Snow and Ice Data Center. Digital Media., Boulder, CO.

Painter, T. H., Dozier, J., Roberts, D. A., Davis, R. E., and Green, R. O. (2003), Retrieval of subpixel snow-covered area and grain size from imaging spectrometer data, *Remote Sensing of Environment*, 85(1): 64-77.

Painter, T. H., and Dozier, J. (2004), Measurements of the hemispherical-directional reflectance of snow at fine spectral and angular resolution, *Journal of Geophysical Research-Atmospheres*, Vol. 109(D18): D18115,10.1029/2003JD004458.

Painter, T. H., Rittger, K., McKenzie, C., Slaughter, P., Davis, R. E., and Dozier, J. (2010a), Assessment of the accuracy of current snow cover mapping algorithms for MODIS, *in preparation*.

Painter, T. H., Rittger, K., McKenzie, C., Slaughter, P., Davis, R. E., and Dozier, J. (2009), Retrieval of subpixel snow covered area, grain size, and albedo from MODIS, *Remote Sensing of Environment*, 113: 868-879, doi: 10.1016/j.rse.2009.01.001.

Painter, T. H., Rosenthal, C. W., Rittger, K., McKenzie, C., Davis, R. E., and Dozier, J. (2010b), Multiple endmember spectral mixture analysis of fractional snow cover from Landsat Thematic Mapper data, *in preparation*.

Powell, R. L., Roberts, D. A., Dennison, P. E., and Hess, L. L. (2007), Sub-pixel mapping of urban land cover using multiple endmember spectral mixture analysis: Manaus, Brazil, *Remote Sensing of Environment*, 106(2): 253-267, doi: 10.1016/j.rse.2006.09.005.

Roberts, D. A., Smith, M. O., and Adams, J. B. (1993), Green vegetation, nonphotosynthetic vegetation, and soils in AVIRIS data, *Remote Sensing of Environment*, 44(2-3): 255-269, doi: 10.1016/0034-4257(93)90020-X.

Roberts, D. A., Gardner, M., Church, R., Ustin, S., Scheer, G., and Green, R. O. (1998a), Mapping chaparral in the Santa Monica Mountains using multiple endmember spectral mixture models, *Remote Sensing of Environment*, 65(3): 267-279.

Roberts, D. A., Gardner, M., Church, R., Ustin, S. L., Scheer, G., and Green, R. O. (1998b), Mapping chaparral in the Santa Monica Mountains using multiple endmember spectral mixture models, *Remote Sensing of Environment*, 65(3): 267-279, doi: 10.1016/S0034-4257(98)00037-6.

Rosenthal, W., and Dozier, J. (1996), Automated mapping of montane snow cover at subpixel resolution from the Landsat Thematic Mapper, *Water Resources Research*, 32(1): 115-130, doi: 10.1029/95WR02718.

Salomonson, V. V., and Appel, I. (2004), Estimating fractional snow cover from MODIS using the normalized difference snow index, *Remote Sensing of Environment*, 89: 351-360.

Salomonson, V. V., and Appel, I. (2006), Development of the Aqua MODIS NDSI fractional snow cover algorithm and validation results, *IEEE Transactions on Geoscience and Remote Sensing*, 44(7): 1747-1756, doi: 10.1109/TGRS.2006.876029.

Schaepman-Strub, G., Schaepman, M. E., Painter, T. H., Dangel, S., and Martonchik, J. V. (2006), Reflectance quantities in optical remote sensing—definitions and case studies, *Remote Sensing of Environment*, 103(1): 27-42, doi: 10.1016/j.rse.2006.03.002.

Stamnes, K., Tsay, S.-C., Wiscombe, W. J., and Jayaweera, K. (1988), Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media, *Applied Optics*, 27: 2502-2509.

Warren, S. G. (1982), Optical properties of snow, *Reviews of Geophysics and Space Physics*, 20(1): 67-89.

Wiscombe, W. J. (1980), Improved Mie scattering algorithms, *Applied Optics*, 19(9): 1505-1509.

APPENDIX 1: COMMON ANCILLARY DATA SETS

1. LAND_MASK_NASA_1KM

a. Data description

Description: Global 1km land/water used for MODIS collection 5
Filename: lw_geo_2001001_v03m.nc
Origin: Created by SSEC/CIMSS based on NASA MODIS collection 5
Size: 890 MB.
Static/Dynamic: Static

b. Interpolation description

The closest point is used for each satellite pixel:

1) Given ancillary grid of large size than satellite grid
2) In Latitude / Longitude space, use the ancillary data closest to the satellite pixel.

2. MDS_L2_CLD_MASK_FILE

a. Data description

Description: MODIS L2 cloud mask 1km
Filename: MOD35_L2.AYYYYDDD.HHMM.005.yyyydddhhmmss.nc /
MYD35_L2.AYYYYDDD.HHMM.005.yyyydddhhmmss.nc.
Where,
MOD35_L2/ MYD35_L2 – Level 2 Cloud Mask from TERRA (MOD) / AQUA (MYD)
A – Nothing to do here
YYYYDDD – 4 digit year plus 3 digit of Julian day
HHMM – 2 digit of hour and 2 digit of minutes in GMT
005 – Processing system version
yyyydddhhmmss – processing date/time
Origin: NASA DAAC
Size: 45 MB
Static/Dynamic: Dynamic

b. Interpolation description

The closest point is used for each satellite pixel:

In Latitude / Longitude space, use the ancillary data closest to the satellite pixel.

APPENDIX 2: DESCRIPTION OF MODIFIED GRAM-SCHMIDT ORTHOGONALIZATION

As described in the Snow Cover Algorithm Theoretical Basis Document, the spectral mixture equation is written as follows:

$$R_{S,\lambda} = \sum_{i=1}^{N} F_i R_{\lambda,i} + \varepsilon_\lambda \quad (1)$$

where F_i is the fraction of endmember i, $R_{\lambda,i}$ is the hemispherical-directional reflectance factor of endmember i at wavelength λ, N is the number of spectral endmembers, and ε_λ is the residual error at λ for the fit of the N endmembers.

This description can be symbolized using matrix structures:

$$A\,x = b \quad (2)$$

where

$$A = \left[\mathbf{u_1} \quad \mathbf{u_2}\right] \text{ (5 by 2 matrix)}, \quad (3)$$

$\mathbf{u_1}$ is the first endmember spectrum (vector of endmember reflectances $R_{\lambda,1}$ as above) and $\mathbf{u_2}$ is the second endmember spectrum (vector of endmember reflectances $R_{\lambda,2}$ as above) for a 3-endmember case (where the third endmember is the photometric shade),

$$x = \left[F_1 \quad F_2\right] \text{ (the spectral fractions above)}, \quad (4)$$

and

$$b = R_{S,\lambda}, \quad (5)$$

the GOES-R ABI spectrum of bands 1, 2, 3, 5, and 6.

Under linear spectral mixture analysis, the spectral fractions are assumed to equal the areal fractions of snow, vegetation, and soil (subject to the shade-normalization step described below).

For example, the endmember matrix for snow directional reflectance with grain size 100 μm and a particular vegetation directional reflectance is given by

$$A = \begin{bmatrix} R_{0.47\mu m,snow_{100\ \mu m}} & R_{0.47\mu m,veg} \\ R_{0.64\mu m,snow_{100\ \mu m}} & R_{0.64\mu m,veg} \\ R_{0.87\mu m,snow_{100\ \mu m}} & R_{0.87\mu m,veg} \\ R_{1.61\mu m,snow_{100\ \mu m}} & R_{1.61\mu m,veg} \\ R_{2.25\mu m,snow_{100\ \mu m}} & R_{2.25\mu m,snow} \end{bmatrix}. \quad (6)$$

The estimate of x comes through the inversion of A if A was square and had linearly independent columns. However, we have an overdetermined case and therefore the solution is not unique.

In the ATBD, we left the description of the matrix operations as the objectively accurate but relatively uninformative statement *the least squares fit arriving at F_i can be solved by several standard methods*. This should have stated specifically that we solve the linear least squares problem (that is, we determine the best fit F_1 and F_2 in this overdetermined case) with the Modified Gram-Schmidt Orthogonalization [*Golub and Van Loan*, 1996]. It can also be solved with methods such as the Singular Value Decomposition or others but we select the robust MGS method. The following description should have found its way (yes, it is more appropriate to speak in the active voice … "We should have inserted the following description ….") into the Snow Cover ATBD. It is through the least squares solution to the overdetermined case that ε_λ appears in the spectral mixture equation. The determination of the matrix inversion is the subject of the remainder of this document.

Note again that A is not invertible, therefore we must minimize:

(7) $$\|b - Ax\|.$$

Assuming that the endmember spectra $\mathbf{u_1}$ and $\mathbf{u_2}$ are linearly independent, we use a QR decomposition with the Modified Gram-Schmidt orthogonalization. The QR decomposition results in a unitary matrix Q (that is, $Q^TQ = I$ (the identity matrix)) and an upper triangular matrix R (do not confuse with reflectance here), such that

(8) $$A = QR$$

and, ultimately,

(9) $$Rx = Q^T b.$$

The matrix Q is determined through the orthogonalization process using a projection operator $\text{proj}_{\mathbf{v}}\mathbf{u}$:

(10) $$proj_{\mathbf{v}}\mathbf{u} = \frac{\langle \mathbf{v},\mathbf{u} \rangle}{\langle \mathbf{v},\mathbf{v} \rangle}\mathbf{v}$$

where <**v**,**u**> indicates the inner product between vectors **v** and **u** and the operator determines the projection of **u** onto **v**. Then the determination of orthogonal basis vectors comes from the series of operations:

$$\mathbf{v}_1 = \mathbf{u}_1$$

(11) $$\mathbf{v}_2 = \mathbf{u}_2 - proj_{\mathbf{v}_1}\mathbf{u}_2$$

M

where $\mathbf{u_1}$ and $\mathbf{u_2}$ are the column vectors of A. The operations can continue but we only model for 2 physical endmembers and 1 photometric shade endmember under the Snow Cover algorithm and as such we leave the operations as used. The orthogonal basis vectors $\mathbf{v_i}$ can now be normalized to provide an orthonormal basis:

$$\mathbf{w}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|}$$

(12) $$\mathbf{w}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|}$$

M

and then

$$Q = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 \end{bmatrix} \tag{13}$$

and

$$R = \begin{pmatrix} \langle \mathbf{w}_1, \mathbf{u}_1 \rangle & \langle \mathbf{w}_1, \mathbf{u}_2 \rangle \\ 0 & \langle \mathbf{w}_2, \mathbf{u}_2 \rangle \end{pmatrix}. \tag{14}$$

The square upper triangular matrix R is invertible and therefore we solve for the spectral fractions according to:

$$x = R^{-1}Rx = R^{-1}Q^T b = \begin{bmatrix} F_1 & F_2 \end{bmatrix}. \tag{15}$$

This calculation then minimizes the metric (7) above with the residual vector that is presented in the spectral mixture equation (1). From the residual vector, we can calculate the spectrum-wide root mean squared error and the shade-normalized snow fraction f_S:

$$f_s = \frac{F_s}{\sum_{p \in s,v,r} F_p} = \frac{F_s}{1 - F_{shade}} \tag{16}$$

where F_s is the snow spectral fraction, F_p are the physical spectral fractions (non-shade), and F_{shade} is the spectral fraction of photometric shade [*Gillespie et al.*, 1990]. Normalizing by the additive complement of the shade fraction accounts for topographic effects on irradiance. The estimates of subpixel vegetation cover, rock cover, and other surface cover are determined with equation (16) as well.

Gillespie, A. R., M. O. Smith, J. B. Adams, S. C. Willis, A. F. Fischer III, and D. E. Sabol (1990), Interpretation of residual images: spectral mixture analysis of AVIRIS images, Owens Valley, California, paper presented at Second Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Workshop, Jet Propulsion Laboratory, Pasadena, CA.

Golub, G. H., and C. F. Van Loan (1996), *Matrix Computations*, 3rd ed., 694 pp., Johns Hopkins University Press, Baltimore.

APPENDIX 3: SOFTWARE DEVELOPMENT DOCUMENTATION

1 INTRODUCTION 1.1 Implementation Concepts: Models and Model Types

As stated above, the FSCA is based on a least squares fit between the mixed spectra observed by the sensor instrument and solar-zenith-angle-specific libraries of *a priori* measured pure endmember spectra. Each of the spectral libraries delivered with SCAG include the spectra for 110 snow endmembers (by grain size radii from 10 to 1,100 microns) and 54 additional spectra for various vegetation, rock/bare-ground and ice/other endmember types.

Given enough bands of input data (and raw processing power) it is technically possible to calculate any desired number of endmember fractions for a given pixel. However, this implementation of the FSCA is constrained by the relative small number of available spectral bands on the GOES-R ABI instrument and the execution-time performance requirements set forth in the MRD. Consequently, this implementation of the FSCA is limited to a maximum of two endmembers.

The easiest and perhaps most important FSCA solution is the pure pixel case (i.e., the entire pixel is covered by a single endmember). Therefore, while we are limited to two endmembers per pixel, we also test the one-endmember solutions.

For practicality's sake, we've grouped the 164 endmembers into four groups: snow, vegetation, rock (including bare ground) and other. While the FSCA keeps track of the specific endmember combinations, the output is generalized into one of these four endmember groups. Working with these limitations the outputs from this implementation are generalized into endmember group combinations as follows:

- Pure snow;
- Pure vegetation;
- Pure rock;
- Pure other;
- Snow fraction plus vegetation fraction;
- Snow fraction plus rock fraction;
- Snow fraction plus other fraction;
- Vegetation fraction plus rock fraction;
- Vegetation fraction plus other fraction and
- Rock fraction plus other fraction.

Currently, “other” is interpreted as ice.

Both physically and mathematically, the FCSA has to account for photometric shade (zero reflectance for each band in the spectrum). For each of the 10 generalized endmember group combinations listed above there is actually an implicit additional endmember (expressed explicitly in the computations, of course) to account for shade. The calculated shade fraction is normalized out of the final solution by dividing the non-shade endmember fractions by one minus the shade fraction.

At this point it is appropriate to introduce the concept of models and model types. In order to calculate the FSC solution we have to compare the spectrum measured by the sensor instrument at the pixel against each possible one and two endmember fractional combinations of the 164 spectral library spectra. These combinations are referred to as models. The model (combination of endmember spectra) yielding the best least squared fit is selected as the optimal solution for that pixel.

The concept of model types addresses two issues. Firstly, as mentioned above, the 164 specific endmembers are generalized into one of four general endmember groups and 10 combinations of those groups. The concept of model types is used to order the large number of possible individual endmember combinations into the 10 generalized endmember group combinations listed above.

The second purpose of model types is to address the issue of model constraints. Only in the very rarest of circumstances will the measured spectrum perfectly match a specific model. Normally there are model residuals to consider. Only when the model residuals and RMSE fall below threshold levels is a model considered acceptable. In addition to grouping and combining models, model types are used to associate varying levels of model fit constraints to each combination of endmember groups. In this FSCA implementation we consider two levels of model fit constraints: loose and tight. The tight constraints are characterized by small acceptable model RMSE and model residual requirements, while the loose constraints relax these requirements.

In the FSCA the model types are ordered by priority as follows:

1. One-endmember model with tight constraints;
2. Two-endmember model with tight constraints;
3. One-endmember model with loose constraints and
4. Two-endmember model with loose constraints.

One should quickly realize that each endmember model has the potential for being calculated twice: once with tight constraints and once with loose constraints.

When retrieving the FSC for a given pixel the algorithm loops through the potential models (endmember combinations) by model type in the order of priority listed above. As mentioned above, the optimal model for a given pixel is that which mostly closely meets the model constraints (smallest RMSE and residuals). In order to limit needless calculations, once we discover an acceptable solution for a given pixel at given priority level we exhaustively examine the remaining models for that priority level but skip all of the models associated with lower priority levels. For instance, if a pixel meets the constraints for priority one, there is no point in examining the models associated with priorities two, three, or four.

1.2 Endmember Memory

The purpose of this section is to describe a key concept inherent in the design of the FSCA as expressed in the SCAG source files.

It's clear that with 165 endmembers and four model types that there are many hundreds of potential endmember models for each pixel. Even when performing a one-time initialization for each model prior to looping through the image pixels, this represents a significant computational burden when one considers that there are millions of pixels in a typical GOES-R ABI image.

One way to reduce this burden is to exploit the fact that the non-snow endmember is relatively (if not entirely) stable. If one could "remember" the non-snow endmembers from image to image, the computational load can be reduced dramatically by limiting the algorithm to only those models that contain that endmember.

This implementation of the FSCA incorporates the concept of endmember memory. Since the non-snow endmember often is not entirely stable (e.g., harvested crop lands) we refresh the memory for each pixel once a day at that pixel's local solar noon.

2 FILES

2.1 Overview

The FSCA is expressed in a program named "scag." There are four types of files associated with the SCAG program, other than the program code itself:

- Configuration files;
- Input files;
- Output files and
- Endmember memory file.

Their purpose, content and format are described in the following sections. The configuration and endmember memory files are of interest primarily to the FSCA programmers. The input files are of interest primarily to the FSCA program operators. And the output files are of interest primarily to the FSCA product users. The following sections will attempt to be sensitive to the needs of these various audiences.

2.2 Configuration Files

2.2.1 Overview

The purpose of this section is to describe the files intrinsic to the design of the FSCA as expressed in the SCAG source code.

The FSCA configuration files contain data that either a) control the flow of the FSCA execution, or b) contain static data required by the FSCA. It is through these files that future modification of many of the performance characteristics associated with the algorithm can be altered without modifying the source code itself. All configuration files are expected by the SCAG program to reside in a single directory. Under no circumstances should these files be altered by the FSCA program operators. Their content not only directs the flow of the FSCA program, but also affects its results. Under certain circumstances, incorrect modification of their content could either a) yield unpredictable erroneous results or b) result in runtime errors. Consequently, the contents of these files should fall within the domain of the FSCA programmers. All configuration parameters that are of interest to the FSCA operators are contained in the program's command-line arguments and can be captured within script files.

2.2.2 SPECLIBS

2.2.2.1 Purpose

The purpose of the SPECLIBS file is to define the configuration of the FSCA's spectral library input files (see #define FILE_SPECLIBS in the SCAG source code). This file is read once by scag to determine the location and characteristics of the spectral libraries available to the FSCA.

2.2.2.2 Content

This file identifies:

- The name of each spectral library file;
- The nominal solar zenith angle for which the file's contents are valid;
- The number of endmember spectra contained in the file;
- The number of spectra wavelengths contained in the file;
- The gain applied to each spectrum value and
- The endmember location in the file for the photometric spectrum.

The gain value is the divisor to be applied to the library spectrum data to align them with the expected range of sensor input data. The current range of library spectrum data is 0 to 10,000. The expected range of sensor input data is 0 to 1,000. Hence the gain value should be set to 10.0 for each entry in the SPECLIBS file. The gain value is closely related to the scag #defined variable DATA_SCALAR. The current value of DATA_SCALAR is 1,000.0, which scales the range of the input sensor data from 0 to 1 to 0 to 1,000. When altering either the gain or the DATA_SCALAR, make sure the other is altered appropriately.

2.2.2.3 Format

The SPECLIBS file contents are ASCII. There is one header record describing the contents of the file. Record fields are space delimited. For example:

(filename) (solar zenith angle) (number of end members) (number of bands)(gain) (shade location)speclib.z30.sli 30 165 7 10.0 0speclib.z45.sli 45 165 7 10.0 0speclib.z60.sli 60 165 7 10.0 0

2.2.3 speclib.z##.sli

Where ## refers to the solar zenith angle.

2.2.3.1 Purpose

The purpose of the speclib.z##.sli files is to store the *a priori* measured solar-zenith-angle-specific pure spectral signature for each possible endmember. These files are read once by scag and held in memory.

2.2.3.2 Content

These files contain:

1. The spectral signature for the photometric shade endmember;
2. The spectral signatures for the snow endmembers by ascending grain size and
3. The spectral signatures for the non-snow endmembers.

All of these files should contain the same number of spectra and bands. The location of endmember spectra should be the same in each file.

The current library files contain spectra for the following wavelengths:

- 0.469 µm;
- 0.555 µm;
- 0.645 µm;
- 0.858 µm;
- 1.240 µm;
- 1.640 µm; and
- 2.130 µm.

2.2.3.3 Format

The speclib.z##.sli files are binary. There are no header records. Each record contains the *a priori* measured spectral signature for one pure endmember. The signatures are stored as four-byte floating-point values, one value for each wavelength.

2.2.4 MODELTYPES

2.2.4.1 Purpose

The purpose of the MODELTYPES file is to define the configuration of the FSCA's workflow by defining the model types (see #define FILE_MODEL_TYPES in the scag source code). This file identifies the files containing the definitions of endmember group combinations, identifies the two and three endmember (including photometric shade) model types and their model file constraint levels, and defines the prioritized order that the model types should be executed. This file is read once by scag to determine the order in which the other FSCA configuration files

should be read into memory and subsequently processed.

2.2.4.2 Content

This file identifies:

1. The name of the model type;
2. The number of endmembers (including photometric shade) in the model type;
3. The number of models defined by the model type;
4. The priority of the model type [from 0 (the highest) to n (the lowest)] and
5. The model type model fit constraint level [0 (tight) or 1 [loose)].

Note that each model type is listed twice: once with a tight constraint level and once with a loose constraint level. Relative to order of priority, for a given model type, the tight constraints should precede the loose constraints. Relative to order of priority, the two-endmember model types should precede the three-endmember model types.

2.2.4.3 Format

The MODELTYPES file contents are ASCII. There is one header record describing the contents of the file. Record fields are space delimited. For example:

(scag model type) (number of end members) (number of models) (sorting weight) (tight/loose, 0/1)basic 2 164 0 0 basic 2 164 2 1 gr 3 660 1 0gr 3 660 3 1met 3 440 1 0 met 3 440 3 1 till 3 220 1 0 till 3 220 3 1 vlc1 3 660 1 0 vlc1 3 660 3 1 vlc2 3 660 1 0 vlc2 3 660 3 1 vlc3 3 330 1 0 vlc3 3 330 3 1 veg1 3 660 1 0 veg1 3 660 3 1veg2 3 660 1 0veg2 3 660 3 1veg3 3 660 1 0veg3 3 660 3 1veg4 3 220 1 0veg4 3 220 3 1gr_veg 3 120 1 0gr_veg 3 120 3 1met_veg 3 80 1 0met_veg 3 80 3 1vlc_veg 3 120 1 0vlc_veg 3 120 3 1lkice 3 330 1 0 lkice 3 330 3 1

2.2.5 #em.name.models

Where # refers to the number of end members and name refers the model type name.

2.2.5.1 Purpose

The purpose of the #em.name.models files is to define the FSCA's retrieval models and the endmember group combination component of the model type definitions (the other component being the model constraints; see section 2.2.6 below). These files are read once by scag and held in memory.

2.2.5.2 Content

These files contain the two-endmember model endmembers and the three-endmember model endmember pairings as part of the model type definitions. Recall that the FSCA model type definitions are comprised of two parts: the combining of endmember groups and the assignment of model fit constraints.

2.2.5.3 Format

The MODELTYPES file contents are ASCII. There are no header records. Record fields are space delimited where needed. For example:

1 or 1 113 2 2 113 3 3 113 .. .. 162 108 116

163 109 116 164 110 116

The numbers are the numerical identifiers for endmembers. The numbers 1 to 110 correspond to the snow endmembers with grain sizes measuring from 1 to 1,100 microns in radii (in 10 micron increments). The remaining identifiers are variously distributed among vegetation, rock, bare ground, and lake ice endmembers.

2.2.6 CONSTRAINTS

2.2.6.1 Purpose

The purpose of the CONSTRAINTS file is to assign model good of fit thresholds to the loose and tight two- and three-endmember model types (see #define FILE_CONSTRAINTS in scag.h). This file is read once by scag held in memory.

2.2.6.2 Content

This file contains:

- The number of endmembers in the model type for which the constraints apply
- The constraint level - either loose or tight
- The minimum and maximum acceptable calculated endmember fraction
- The threshold maximum model RMSE
- The threshold maximum residual for a spectral signature band
- The maximum number of successive wavelength-ordered bands that are allowed to exceed the threshold maximum residual.

If, for any model, any of these constraints are exceeded, that model is rejected. The accepted model is that which meets these constraints with the minimum RMSE after all models within its priority level have been tested. Once all models within a priority level have been tested **and** an acceptable model has been identified, further modeling for a pixel is unnecessary (i.e., lower priority models are ignored).

2.2.6.3 Format

The MODELTYPES file contents are ASCII. There is one header record describing the contents of the file. Record fields are comma and space delimited. For example:

(num of EMs) (tight|loose, 0|1) (frac min/zero/max pairs, 1pr / EM) (rmsthresh) (rthresh) (rcount)2, 1, -0.01 0.10 1.01, -0.01 0.10 1.01, 30.0, 30.0, 32, 0, -0.01 0.10 1.01, -0.01 0.10 1.01, 15.0, 15.0, 33, 1, -1.01 0.10 2.01, -1.01 0.10 2.01, -1.01 0.10 2.01, 50.0, 50.0, 33, 0, -0.01 0.10 1.01, -0.01 0.10 1.01, -0.01 0.10 1.01, 15.0, 15.0, 3

2.2.7 EMTYPES

2.2.7.1 Purpose

The purpose of the EMTYPES file is to assign endmembers to an endmember group (see #define FILE_EMTYPES in the SCAG source code). This file is read once by scag held in memory. There are only 164 entries in EMTYPES because these are the only physical endmembers. The photometric shade holds position ‘0’ in each spectral library and has no emtype –

arbitrary and perhaps counterintuitive, but it is what we've done for many years. The first 110 endmembers are modeled snow spectra. The next 3 are measurements of vegetation-shaded snow – hence they are given emtype as snow. However, because of the complexity of use of these endmembers, they are not included in the model.

2.2.7.2 Content

This file contains the endmember group associated with each endmember as follows

0: Snow
1: Vegetation
2: Rock/bare ground
3: Other (lake ice)

2.2.7.3 Format

The EMTYPES file contents are ASCII. There are no header records. Each record contains a single value, which is the endmember group identifier (0-3) as described above. The first record identifies the endmember group identifier for endmember one. The second record identifies the endmember group identifier for endmember two. And so on.

2.2.8 GSTABLE

2.2.8.1 Purpose

The purpose of the GSTABLE (Grain Size Table) file is to assign a snow grain size radius (in microns) to each endmember (see #define FILE_GSTABLE in scag.h). This file is read once by scag held in memory.

2.2.8.2 Content

This file defines the snow grain size radius (in radians) for each endmember. This file differs from many of the other configuration files in that it includes an entry for the photometric shade endmember.

2.2.8.3 Format

The GSTABLE file contents are ASCII. There are no header records. Each record contains a single value, which is the snow grain size (radius in microns) for a particular endmember. The first record defines the grain size for photometric shade and should have a value of zero. The next 110 records contain the grain sizes for the snow endmembers and should range, in increasing value, from 10 μm to 1,100 μm. The remaining records define the snow grain sizes for the remaining endmembers (all non-snow) and should have a value of zero. Position 0 is shade and has grain size of 0. The next 110 endmembers are snow and they have grain sizes ranging from 10 to 1100 micrometers. The next 54 endmembers are non-snow endmembers (vegetation, soils) and therefore have no associated defined grain size (at least, in the sense that interests us).

2.3 Input Files

2.3.1 Overview

All of the non-configuration inputs to the FSCA are of a single basic format. While the contents of the files may vary in data type, dimension and spatial and temporal location (as expressed in their header files), these differences are handled by scag's file I/O subsystem. This modular subsystem, as it currently exists, is designed to handle the file format described in this section. However, due to its modular nature, the I/O subsystem can easily be extended to accommodate additional formats.

The current inputs to scag are made up of two components: a binary header file and a flat binary raster data file. The format of each file is described immediately below. Specific file contents are the topic of subsequent subsections.

2.3.2 File Formats

2.3.2.1 Header File Format

Two header files are produced for each set of output data, a binary header and an ASCII header. Only the binary header file is required of input files.

The contents of the binary header files (filename extension of ".hdr") are as follows:

Table 1. Header file format

The data type values and their respective types are as follows:

Byte Offset	Data Type	Value	Default	Required
0	Unsigned char	Data type		•
1	double float	Data scalar	1.0	•
9	double float	Data intercept	0.0	•
17	double float	Minimum data value	DBL_MAX	
25	double float	Maximum data value	-DBL_MAX	
33	double float	No-data value		•
41	double float	X-axis resolution	0.0	
49	Unsigned long	Number of file columns		•
53	Unsigned long	Number of bytes per row		•
57	double float	Minimum x-axis coordinate	DBL_MAX	
65	double float	Maximum x-axis coordinate	-DBL_MAX	
73	double float	Y-axis resolution	0.0	
81	Unsigned long	Number or file rows		•
85	double float	Minimum y-axis coordinate	DBL_MAX	
93	double float	Maximum y-axis coordinate	-DBL_MAX	
101	double float	Data nominal time		•
109	double float	Data minimum time	DBL_MAX	•
117	double float	Data maximum time	-DBL_MAX	•

1: signed char
2: double float
3: float
4: long
5: short
6: unsigned char

7: unsigned long
8: unsigned short

Data scalar and intercept can be used to compress the data:

actual_value = data_intercept + file_value * data_scalar

One should note that the no-data value is as it exists in the file, not the uncompressed actual value.

Other than the extension, the header file should have the same name as the data file. Header files must have a .hdr extension.

An ASCII version of this header data is produced and stored in a file with a .hda extension. The data is a plain-text representation of the binary data and is provided for convenience in interpreting the data without one's decoding the binary header. It can also be used directly or modified slightly for use in data visualization software.

2.3.2.2 Data File Format

The data files are flat binary files in which the pixels are arranged in columns by rows as defined in the accompanying header file (see section 2.3.2.1 above). The size of the file is dictated by the data type, number of columns and number of rows.

Other than the extension, the data file should have the same name as the header file. Data files must have a .dat extension.

2.3.3 Surface Reflectance Data

The FSCA program (scag) is currently configured to retrieve FSC from GOES-R ABI surface reflectance values. Future modifications to the FSCA will employ two additional bands from the thermal range of the electromagnetic (EM) spectrum to assist in the discrimination of snow from playa in desert regions (see the FSCA ATBD and Table 1 above).

The scag program has two operating modes (Table 3), each of which requires MODIS surface reflectance data as inputs. In the first mode, the operations or GOES-R ABI simulation mode, five bands of MODIS data are used as proxy data to simulate FSC retrievals from GOES-R data. In the second mode, the validation mode, seven bands of MODIS data are used to validate the simulated ABI retrievals against the FSC retrieval performance measures known, through rigorous experimentation and validation, for the MODIS instrument (see the FSCA ATBD).

MODIS surface reflectance data are available from two sources:

1 The MODIS MOD09 product (several days of latency) or
2 The MODIS MOD09_SPA program which uses near real-time MOD03, MOD02_QKM, MOD02_HKM, MOD02_1KM and NCEP GDAS data as inputs to a radiative transfer model.

The surface reflectance values recorded in these files are floating point fractional values ranging from zero to one. They are scaled to become aligned with the spectral library files (see #define DATA_SCALAR in scag.h).

Table 2. Surface reflectance bands for GOES-R and MODIS and their use.

ABI Band	ABI Band Wavelength	MODIS Band	MODIS Band Wavelength	Operations Mode	Validation Mode
1	0.45–0.49 μm	3	0.46–0.48 μm	•	•
		4	0.54–0.56 μm		•
2	0.59–0.69 μm	1	0.62–0.67 μm	•	•
3	0.85–0.88 μm	2	0.84–0.88 μm	•	•
		5	1.23–1.25 μm		•
5	1.58–1.64 μm	6	1.63–1.65 μm	•	•
6	2.22–2.28 μm	7	2.10–2.16 μm	•	•

2.3.4 Land-Water Mask

The land-water mask is used by the FSCA to restrict FSC retrievals to land pixels. The mask is used as a filter. Currently, the FSCA does not process pixels which are over ocean (see #define DEEP_OCEAN_VALUE, #define MODERATE_OCEAN_VALUE, and #define SHALLOW_OCEAN_VALUE in scag.h)

Future versions of the FSCA may allow the algorithm to process ocean pixels.

The land-water mask values are interpreted as follows:

0: Shallow ocean
1: Land
2: Coast/shore lines
3: Shallow inland water
4: Intermittent/ephemeral water
5: Deep inland water
6: Moderate/continental ocean
7: Deep ocean

2.3.5 Cloud Mask

The cloud mask is used by the FSCA to qualify the algorithm's grain size dependent cloud retrievals in its quality output file (see section 2.4.10 below). The FSCA does not mask out clouds. Rather, it takes advantage of the fact that clouds are actually snow crystals of very small radii suspended in the atmosphere. As such, from the FSCA perspective, snow retrievals below a certain grain size threshold are mapped as clouds (see #define CLOUD_GRAIN_SIZE in scag.h).

The FSCA cloud retrievals are indexed through the cloud mask and reported in the algorithm's quality output file.

The cloud mask currently employed by the FSCA is the first byte of the MODIS MOD35_L2 product. The first bit in the byte, bit zero, indicates whether or not cloud cover was determined for a pixel: 0 – cloud cover undetermined, 1 – cloud cover determined. The next two bits' values, bits one and two, are interpreted as follows:

0: Confident cloudy
1: Probably cloudy
2: Probably clear
3: Confident clear

2.3.6 Solar Zenith Angle

The solar zenith angle file is used by the FSCA to select the correct spectral signature library for a given pixel's FSC retrieval. And to filter out pixels that are either too bright or too dim for processing (see #define MAX_SOLAR_ZENITH_ANGLE and #define MIN_SOLAR_ZENITH_ANGLE in scag.h). The solar zenith angle ranges from zero degrees (the sun is immediately above) to 90 degrees (the sun is on the horizon) to 180 degrees (the sun is below the horizon). The currently acceptable range of solar zenith angles is 0 to 67.5 degrees.

2.3.7 Sensor Zenith Angle

Similar to solar zenith angle, the sensor zenith angle (view angle) is a measure of the GOES-R ABI sensor position relative to a given pixel. A value of zero indicates that the sensor is immediately above. The sensor zenith angle is used by the FSCA to filter out pixels that are too far away from nadir (see #define MAX_SENSOR_ZENITH_ANGLE and #define MIN_SENSOR_ZENITH_ANGLE in scag.h). These limb pixels, due to their position relative to the instrument, are distorted beyond usefulness. The currently acceptable range of sensor zenith angles is 0 to 55 degrees of arc.

2.3.8 Longitude

The longitude file contains the longitudinal position of each pixel at datum. It does not account for distortions due to parallax. The longitude file is used to keep track of non-snow endmember "memory" (see section 3.5 below).

2.3.9 Latitude

The latitude file contains the latitudinal position of each pixel at datum. It does not account for distortions due to parallax. The latitude file is used to keep track of non-snow endmember "memory" (see section 2.5 below).

2.4 Output Files

2.4.1 Overview

All of the outputs from the FSCA are of a single basic format. While the contents of the files may vary in data type, dimension and spatial and temporal location (as expressed in their

header files), these differences are handled by scag's file I/O subsystem. This modular subsystem, as it currently exists, is designed to handle the file format described above (see section 2.3.2 above). However, due to its modular nature, the I/O subsystem can easily be extended to accommodate additional formats.

The current outputs to scag are made up of three components: a binary header file (ending with ".hdr"), an ASCII header file (ending with ".hda"), and a flat binary raster data file. The format of each file is described above. Specific file contents are the topic of subsequent subsections.

2.4.2 File Formats

The output file format is the same as the input file format (see section 2.3.2 above). The data type for each output file is float. The universal no data value, largely for display purposes, is
0.01 (see #define NDV in scag.h).

2.4.3 Snow Fraction

The snow endmember fraction retrieved by the FSCA. The range in values is 0.0 to 1.0. The sum of fractions (snow, vegetation, rock and other) should not exceed 1.0. If below a specified grain size threshold (currently a radius of 40 microns; see #define CLOUD_GRAIN_SIZE_THRESHOLD in scag.h) the retrieval can be assumed to be cloud rather than snow (see the FSCA ATBD).

The strings "snow_fraction.hdr", "snow_fraction.hda", and "snow_fraction.dat" for the header and data files will be appended to the output file base pathname provided by the operator.

2.4.4 Vegetation Fraction

The vegetation endmember fraction retrieved by the FSCA. The range in values is 0.0 to 1.0. The sum of fractions (snow, vegetation, rock and other) should not exceed 1.0.

The strings "vegetation_fraction.hdr", "vegetation_fraction.hda", and "vegetation_fraction.dat" for the header and data files, respectively, will be appended to the output file base pathname provided by the operator.

2.4.5 Rock Fraction

The rock and bare ground endmember fraction retrieved by the FSCA. The range in values is 0.0 to 1.0. The sum of fractions (snow, vegetation, rock and other) should not exceed 1.0.

The strings “rock_fraction.hdr”, “rock_fraction.hdr”, and “rock_fraction.dat” for the header and data files will be appended to the output file base pathname provided by the operator.

2.4.6 Other Fraction

The “other” endmember fraction (not snow, vegetation or rock; in this case, it is lake ice) retrieved by the FSCA. The range in values is 0.0 to 1.0. The sum of fractions (snow, vegetation, rock and other) should not exceed 1.0.

The strings “other_fraction.hdr”, “other_fraction.hda” and “other_fraction.dat” for the header and data files will be appended to the output file base pathname provided by the operator.

The pixel has been identified as existing in inland water, a further investigation of the snow and other (ice) fractions is done. If no snow nor ice exists on such a pixel, it is coded as water, with an appropriate setting of the pixel's snow value (0.0) in the snow fraction data file and the pixel's quality value in quality.dat

2.4.7 Shade Fraction

The photometric shade endmember fraction retrieved by the FSCA. The range in values is 0.0 to 1.0. The shade fraction is used to normalize the non-shade endmembers by dividing their fractions by one minus the shade fraction. In essence the shade fraction is proportionally distributed among the non-shade endmembers.

The strings “shade_fraction.hdr”, “shade_fraction.hda”, and “shade_fraction.dat” for the header and data files will be appended to the output file base pathname provided by the operator.

2.4.8 Snow Grain Size

This is the grain size radii (in microns) for the snow endmembers. Snow which has a grain size falling below a specified gain size threshold (currently a radius of 40 microns; see #define CLOUD_GRAIN_SIZE_THRESHOLD in the source code) can be assumed to be cloud rather than snow (see the FSCA ATBD).

The strings “grain_size.hdr”, “grain_size.hda”, and “grain_size.dat” for the header and data files will be appended to the output file base pathname provided by the operator.

2.4.9 Binary snow

With the calculation of a fractional snow cover, one can define a threshold (currently a fraction of 0.10; see #define BINARY_SNOW_THRESHOLD), below which one might consider a pixel to be snow free; above that threshold, the pixel might be considered to have snow within it.

The strings “binary_snow.hdr”, “binary_snow.hda”, and “binary_snow.dat” for the header and data files will be appended to the output file base pathname provided by the operator.

2.4.10 RMS

The calculated root mean squared error for the FSCA retrievals.

The strings “rms.hdr”, “rms.hda”, and “rms.dat” for the header and data files will be appended to the output file base pathname provided by the operator.

2.4.11 Quality

This file contains a quality value associated with each FSCA retrieval.

> 0.00: if the no data value was encountered in any of the inputs, or
> 1.00: if the pixel falls on water, or
> 2.00: if the solar zenith angle is too small, or
> 3.00: if the solar zenith angle is too large, or
> 4.00: if the pixel could not be modeled, or
> 5.00: if the pixel had bad source data, or
> 6.00: if the pixel has bad horizontal location (latitude or longitude) metadata, or
> 7.00: if the pixel has an unreasonable sensor angle value, or
> 8.00: if the pixel has an unreasonable solar zenith angle, or
> 10.00: if the pixel could be modeled but is snow free, or 2f.ff: if the pixel could be modeled and has snow (where fife is the snow fraction), or 3f.ff: if the pixel could be modeled and has shaded snow (zero grain size) (should be physically impossible but is mathematically feasible) (where f.ff is the snow fraction), or 4f.ff: if the pixel could be modeled and has cloud by grain size (where f.ff is the cloud fraction).

The flags may be modified by the FSCA for cloud masking as follows:

> Add 100.00 if cloud masked, or Add 200.00 if probably cloud masked, or Add 300.00 the cloud mask is undetermined,

The flags may be modified by the FSCA for the following second-most prominent non-snow endmember in a modeled pixel:

> Add 1,000.00 if the most prominent non-snow endmember is vegetation, or Add 2,000.00 if the most prominent non-snow endmember is vegetation, or Add 8,000.00 if the most prominent non-snow endmember is other, or Add 9,000.00 if the most prominent non-snow endmember is photometric shade

The flags may be modified by the FSCA for the following condition:

> Add 10,000 if the pixel was modeled but beyond the sensor angle threshold (See the following in the scag.h source code:

#define BAD_DATA_FLAG #define BAD_LATLON_FLAG #define BAD_SENSOR_FLAG #define BAD_SOLAR_FLAG #define CLOUD_MASKED_FLAG;#define CLOUD_MASK_UNDETERMINED_FLAG #define MODELED_CLOUD_BY_GRAIN_SIZE_FLAG #define MODELED_SHADED_SNOW_FLAG #define MODELED_SNOW_FLAG #define MODELED_SNOW_FREE_FLAG #define NDV_FLAG #define NOT_MODELED_FLAG #define PROBABLY_CLOUD_MASKED_FLAG #define SECOND_EM_OTHER_VALUE #define SECOND_EM_ROCK_VALUE #define SECOND_EM_SHADE_VALUE #define SECOND_EM_VEG_VALUE #define TOO_BRIGHT_FLAG #define TOO_DARK_FLAG #define TOO_FAR_FLAG #define WATER_FLAG.)

The strings “quality.hdr”, “quality.hda”, and “quality.dat“ for the header and data files will be appended to the output file base pathname provided by the operator.

2.4.12 Quality Flags

Along with the floating-point quality values as described in section 2.4.10, a one-byte word is used to compactly hold quality information about a pixel. A value of zero for the byte indicates that modeling was completed on the corresponding pixel (independent of the pixel's containing snow). The "turning on" of a particular bit indicates the reason why the pixel was not modeled. Below is a list of the bits and their corresponding meanings

0: (least-significant bit): no-data value in band data,
1: missing data in band data,
2: modeled cloudy,
3: water,
4: solar zenith angle less than 0 or greater than MAX_SOLAR_ZENITH_ANGLE,
5: sensor zenith angle less than 0.0 or greater than MAX_SENSOR_ZENITH_ANGLE,
6: bad metadata or ancillary data, and
7: (most significant bit) : other reason

The strings “quality_bits.hdr”, “quality_bits.hda”, and “quality_bits.dat” for the header and data files will be appended to the output file base pathname provided by the operator.

2.5 Endmember Memory File

2.5.1 Overview

The endmember memory file keeps track of the last endmember combination observed for longitude-latitude coordinate pairs. Image-to-image memory of last observed endmembers allows the FSCA to execute very rapidly by avoiding needless calculations associated with pointless retrieval models. This is possible by taking advantage of the fact that the non-snow endmember is often stationary, or nearly so.

Needless to say, it is important that the endmember memory file not be deleted or altered in any way.

2.5.2 File Format

The endmember memory file is a header-less flat binary file containing endmember combination data at evenly spaced latitude-longitude coordinate pair locations (rows by columns). The file’s dimensions are defined by the following self-documenting variables in

scag.h:

> #define MAP_RESOLUTION_SECONDS;#define MAP_HEIGHT_DEGREES;#define MAP_NDV;#define MAP_WIDTH_DEGREES;#define MAP_X_ORIGIN (upper left-hand corner; pixel center);#define MAP_Y_ORIGIN (upper left-hand corner; pixel center) and#define SNOW_ENDMEMBER

One should select a map resolution that can accommodate the minimum pixel spacing of the unevenly spaced image pixels. Otherwise multiple image pixels may be mapped to the same map pixel.

The SNOW_ENDMEMBER variable is a convenience variable that is used to generalize all of the many snow endmembers (they vary by grain size; see section 2.2.8 above) into a single endmember ID value in the map. Make sure that this variable is set to a value that does not conflict with the non-snow endmembers.

Four one-byte, interleaved values are stored at each map location:

1 The solar zenith angle to the nearest degree;
2 The sensor zenith angle to the nearest degree;
3 The first endmember ID. If snow is a model endmember it will be identified in this byte and
4 The second endmember ID. If the best-fitting model is a two-endmember (including photometric shade) "pure" model, both the first and second endmember ID will be set to the same value.

3 PRACTICAL CONSIDERATIONS

3.1 Overview

The purpose of this section is to provide an overview of the issues involved in implementing the FSCA from a software perspective. Please refer to the FSCA ATBD and Validation Plan for in depth discussions of all of the FSCA's practical considerations.

3.2 Numerical Computation Considerations

The FSCA relies on two primary inputs:

1 Surface reflectance values for GOES-R ABI channels 1, 2, 3, 5 and 6; and
2 An *a priori* calculated cloud mask.

Some could argue that snow, from a spectral analysis perspective, behaves like a low-altitude, large-particle cloud. It is possible to leverage the spectral mixture analysis for FSC and the rapid GOES-R image acquisition schedule to assist the AWG Cloud Team's ACM product. Temporal signatures of grain size can provide an additional capability for cloud masking as short terms changes in grain size indicate cloud presence. Since the FSCA (which depends on a cloud mask) and the cloud mask algorithm (which depends on a snow mask) are interdependent, it seems reasonable to pursue synergies between the two algorithms.

FSCA as expressed in GOESRSCAG relies on linear transformations to decompose a pixel's spectral signature into its constituent spectra. GOESRSCAG employs (or will employ) several strategies to reduce its computational load:

1 Limit the number of possible endmembers to two (snow and a single non-snow endmember from a limited list of possibilities) plus shade. GOESRSCAG retrieves, for example, combinations of snow and vegetation or snow and bare ground. But it does not retrieve combinations of snow and vegetation and bare ground. Since the primary aim is the determination of FSC (as opposed to the identity and proportion of the non-snow constituents), this limit minimizes the dimensions of the matrices (Status: Done);

2 Limit the number of possible snow grain size spectra (Status: Done);

3 Limit the number of spectra within a possible non-snow endmember spectrum library (Status: Done);

4 Integrate (and then optimize) spectrum mixing analysis modeling with the final endmember sorting/selection logic execute a FSC retrieval with as few matrix operations as possible (Status: Done) and

5 Buffer repetitive, intermediate calculations between time sequential GOES-R ABI images (e.g., the identity of the non-snow endmember for a given pixel) (Status: Done)

3.3 Programming and Procedural Considerations

The GOESRSCAG FSCA is a pixel-by-pixel algorithm that could benefit by running on time-sequential images (see section 3.2 above). The algorithm is reliant on matrix operations that impact program design and implementation. While this reliance impacts programming considerations, we addressed strategies for its mitigation in section 3.2 above.

Additionally, we recognize that the GOES-R satellite will not be deployed for several years. With that in mind, rather than adopting a more traditional single-threaded architecture, we elected to implement the FSCA as a multi-threaded application more suitable to the multiprocessor computers anticipated for the near future.

3.4 Quality Assessment and Diagnostics

The following procedures are recommended for diagnosing the performance and in, some cases, real time validation of the FSCA:

1 Monitor the percentage of specific endmembers in regional areas where these values should be nearly constant after snow cover reaches 100 percent during accumulation in the fall or snow cover reaches 0 percent during ablation in the spring;

2 Assess persistence/consistency of fractional snow cover by pixel. There should be no rapid oscillations in FSC and grain size for a given pixel. FSC for a given pixel should vary smoothly in the temporal domain;

3 Assess errors of confusion between cloud cover and snow cover;

4 Assess fractional snow cover retrievals with high spatial resolution, polar-orbiting sensors such as the Landsat Thematic Mapper;

5 Assess fractional snow cover retrievals with physically based, energy- and mass-balanced snow models;

6 Periodically review the individual test results to look for artifacts or non-physical behaviors;

7 Maintain a close collaboration with the other teams using the FSC in their product generation and

8 Maintain a close collaboration with the cloud teams to resolve issues associated with snow/cloud discrimination.

3.5 Exception Handling

The FSCA will include checking the validity of each required channel before executing its retrievals. FSCA also expects the Level 1b processing to flag any pixels with missing spectral data, geolocation or viewing geometry information.

The following additional pixel-by-pixel exceptions will be identified and flagged by the FSCA in its output:

- Missing data in any of the input files;
- Clouds identified by grain size and/or the cloud mask algorithm;
- Water identified by the water mask;
- Pixels below the solar zenith angle threshold;
- Pixels which are saturated;
- Pixels missing surface reflectance RTM correction and
- Pixels too close to limb.

In these cases an appropriate flag will be set to indicate that no FSC retrieval was made for that pixel (see section 2.4.10 and 2.4.11 above).

4 PROGRAMMER'S REFERENCE

4.1 Overview

The FSCA is expressed in a binary executable program named scag. For a cursory review of scag's purpose and implementation please refer to the introduction to this document. For a more detailed review of the physical and mathematical basis for scag one is encouraged to read the FSCA ATBD and it's listed references. An excellent introduction to the concepts captured in the FSCA is provided in *A Survey of Spectral Unmixing Algorithms* (Keshava, 2003).

The basic outline for the scag program is illustrated in the flowchart depicted in figure 8. The remainder of the section will devoted to describing, in detail, the implementation of the FSCA in the scag source code.

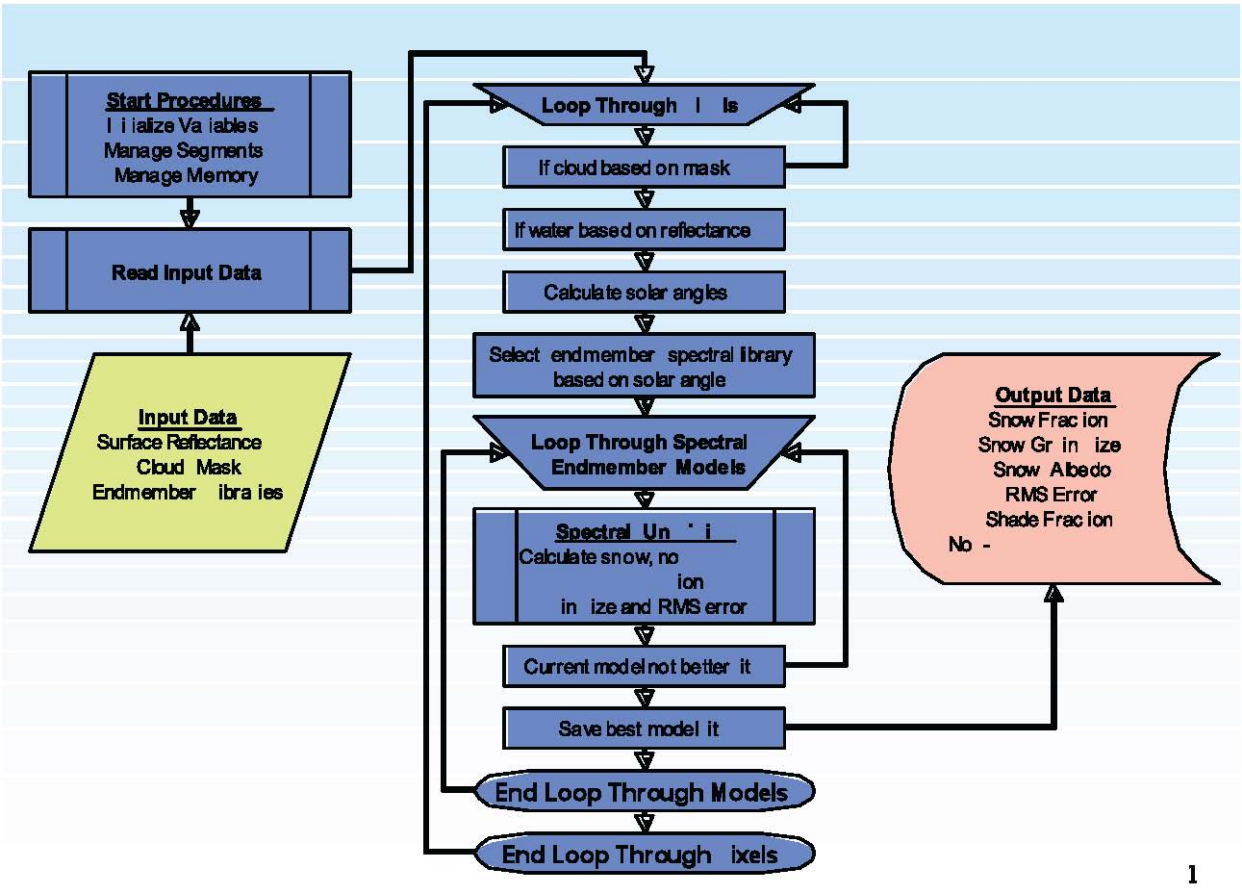


Figure 1. Basic flowchart of core FSCA elements implemented in scag

4.2 Programming Language

The scag program is written in POSIX C using declared variable types intended for a 32-bit compilation. This program can be compiled and executed on a 64-bit computer. However, its makefile must first be edited to include the –m32 compiler flag (see section 4.3 below).

The FSCA is heavily dependent on algorithmic solutions involving repeated matrix cross products. While several strategies for mitigating this computational burden are incorporated into scag, one in particular has an impact on programming language considerations. Namely, scag is designed for multithreading. The scag program's main thread orchestrates all I/O, while each of the remaining threads calculates FSC retrievals at the pixel level (each thread being responsible for all of the calculations required to retrieve FSC for its current pixel).

4.3 Makefile

All of scag's source code is divided into one file per function, and one function per file. Consequently, scag's is relatively straightforward and contains the following lines:

```
CC=gcc CFLAGS=-Wall -Wshadow -Wpointer-arith -ansi -pedantic -D_REENTRANT -O3LDFLAGS=-lm -
lpthreadOBJECTS=fit_constraints.o \
```

```
        fraction.o \       initialize_models.o error_initialize_models.o \       input_file_io.o error_input_file_io.o
    \       main.o error_main.c \       mixture.o \       model_types_io.o error_model_types_io.o \
    next_blank.o \       next_comma.o \       next_field.o \       output_file_io.o error_output_file_io.o \
    process_thread.o error_process_thread.o \       shade_normalization.o \       signal_handler.o \
    spectral_library_io.o error_spectral_library_io.o \help.o

scag : $(OBJECTS)       $(CC) $(CFLAGS) $(LDFLAGS) $(OBJECTS) -o $@

clean :       rm -f *.o scag
```

For 64-bit compilation add the –m32 flag at the end of CFLAGS.

4.4 Included Files

The following files are included in the header file, scag.h:

> #include <errno.h> #include <fcntl.h> #include <getopt.h>#include <math.h> #include <pthread.h>#include <signal.h>#include <stdio.h> #include <stdlib.h> #include <string.h>#include <sys/stat.h>#include <sys/types.h>#include <unistd.h> #include <values.h>.

4.5 Variable Definition Macros

The following variable definition macros are defined in scag.h. Each global variable definition is prepended with #define:

- BAD_DATA_FLAG 5.0 : The quality flag for bad source data;
- BAD_LATLON_FLAG 6.0 : The quality flag for bad latitude or longitude data;
- BAD_SENSOR_FLAG 7.0 : The quality flag for bad sensor angle data;
- BAD_SOLAR_FLAG 8.0 : The quality flag for bad solar zenith angle data;
- BINARY_SNOW_NDV -1 : The no-data value;
- BINARY_SNOW_THRESHOLD 0.10 : The threshold snow fraction for snow coverage. Above this fraction, snow covers the pixel; below it, the pixel is considered snow-free;
- CLOUD_GRAIN_SIZE_THRESHOLD 40.0 : The threshold grain size radius for cloud in microns. A snow grain size of at least this radius can be considered cloud ;
- CLOUD_MASKED_FLAG 100.0 : Value added to the FSCA quality flag to indicate definite ACMA clouds;
- CLOUD_MASK_UNDETERMINED_FLAG 300.0 : Value added to the FSCA quality flag to indicate that the FSCA was unable to determine clouds;
- DATA_SCALAR 1000.0 : Scalar applied to surface reflectance data align them with the spectral library data;
- DEEP_INLAND_WATER_VALUE 5.0 : Deep-ocean water value in the land-water mask;
- DEEP_OCEAN_VALUE 7.0 : Deep-ocean water value in the land-water mask;
- EXIT_BAD 2 : Program exit value indicating bad execution;
- EXIT_GOOD 0 : Program exit value indicating good execution;
- EXIT_HELP 1 : Program exit value indicating the help information was shown;
- EXIT_INTERRUPTED 9 : Program exit value indicating program was interrupted;
- FILE_CONSTRAINTS "CONSTRAINTS" : Name of the FSCA constraints file;
- FILE_EMTYPES "EMTYPES" : Name of the FSCA endmember types file;
- FILE_GSTABLE "GSTABLE" : Name of the FSCA snow grain size file;
- FILE_MODELTYPES "MODELTYPES" : Name of the FSCA model types file;
- FILE_SPECLIBS "SPECLIBS" : Name of the FSCA spectral library configuration file;

- INTERMITTENT_INLAND_WATER_VALUE : Intermittent-water value in the land-water mask;
- MAP_RESOLUTION_SECONDS 30.0 : Resolution of the endmember memory file in seconds;
- MAP_HEIGHT_DEGREES 49.0 : Height of the endmember memory file in degrees;
- MAP_NDV UCHAR_MAX : Endmember memory file no-data value;
- MAP_WIDTH_DEGREES 103.0 : Width of the endmember memory file in degrees of longitude;
- MAP_X_ORIGIN -168.0 : Upper left-hand corner x-axis origin of the endmember memory file (pixel center) in longitude;
- MAP_Y_ORIGIN 71.0 : Upper left-hand corner y-axis origin of the endmember memory file (pixel center) in latitude;
- MAX_CONFIG_PATH_LENGTH 245 : Maximum command line length for the FSCA configuration file directory name;
- MAX_INPUT_PATH_LENGTH 252 : Maximum command line length for the FSCA input file names;
- MAX_MAP_PATH_LENGTH 256 : Maximum command line length for the FSCA endmember memory file name;
- MAX_OUTPUT_PATH_LENGTH 233 : Maximum command line length for the FSCA output file names;
- MAX_SENSOR_ZENITH_ANGLE 85.0 : Maximum acceptable sensor view angle;
- MAX_SOLAR_ZENITH_ANGLE 85.5 : Maximum acceptable solar zenith angle;
- MIN_SOLAR_ZENITH_ANGLE 0.0 : Minimum acceptable solar zenith angle;
- MODELED_CLOUD_BY_GRAIN_SIZE_FLAG 40.0 : FSCA quality flag value indicating clouds were mapped by grain size;
- MODELED_FLAG 0.0 : Initial FSCA quality flag value;
- MODELED_SHADED_SNOW_FLAG 30.0 : FSCA quality flag value indicating shaded snow was mapped;
- MODELED_SNOW_FLAG 20.0 : FSCA quality flag value indicating snow was mapped;
- MODELED_SNOW_FREE_FLAG 10.0 : FSCA quality flag indicating that no snow nor clouds was mapped;
- MODERATE_OCEAN_VALUE 6.0 : Moderately-deep (continental shelf) ocean water value in the land-water mask;
- NDV -0.01 : Universal output no-data value;
- NDV_FLAG 0.0 : FSCA quality flag indicating that an input no data value was encountered;
- NOT_MODELED_FLAG 4.0 : FSCA quality flag indicating the pixel was not modeled;
- PROBABLY_CLOUD_MASKED_FLAG 200.0 : Value added to the FSCA quality flag to indicate probable FSCA clouds;
- PROGRAM "scag" : Program name;
- SECOND_EM_OTHER_VALUE 8000.0 : Value added to the FSCA quality flag to indicate the most prominent nonsnow endmember was something else;
- QUALITY_BIT_BAND_DATA 2 : Value in quality bits field to indicate bad band (raw) data;
- QUALITY_BIT_META_NDV 1 : Value in quality bits field to indicate missing metadata;
- QUALITY_BIT_CLOUD 4 : Value in quality bits field to indicate clouds;
- QUALITY_BIT_LALO 64 : Value in quality bits field to indicate the pixel is outside of the bounding box;
- QUALITY_BIT_MODELED 0 : Value in quality bits field to indicate the pixel was modeled;
- QUALITY_BIT_OTHER 128 : Value in quality bits field to indicate some other reason why the pixel was not modeled;
- QUALITY_BIT_SENSOR 32 : Value in quality bits field to indicate the pixel is beyond the allowed sensor zenith angle;

- QUALITY_BIT_SOLAR 16 : Value in quality bits field to indicate pixel is beyond allowed solar zenith angle;
- QUALITY_BIT_WATER 8 : Value in quality bits field to indicate pixel is over (salt) water;
- RETURN_BAD 0 : Function return value indicating bad execution;
- RETURN_GOOD 1 : Function return value indicating good execution;
- SECOND_EM_ROCK_VALUE 2000.0 : Value added to the FSCA flag to indicate the most prominent nonsnow endmember was rock;
- SECOND_EM_SHADE_VALUE 9000.0 : Value added to the FSCA quality flag to indicate the most prominent nonsnow endmember was shade;
- SECOND_EM_VEG_VALUE 1000.0 : Value added to the FSCA quality flag to indicate the most prominent nonsnow endmember was vegetation;
- SHALLOW_INLAND_WATER_VALUE 3.0 : Shallow ocean water value in the land-water mask;
- SHALLOW_OCEAN_VALUE 0.0 : Shallow ocean water value in the land-water mask;
- SNOW_ENDMEMBER 0 : Endmember memory file snow endmember value;
- THREAD_WAIT_SECONDS 60 : Number of seconds to wait on thread before assuming thread problem;
- TOO_BRIGHT_FLAG 2.0 : FSCA quality flag indicating that the pixel is too bright;
- TOO_DARK_FLAG 3.0 : FSCA quality flag indicating that the pixel is too dark;
- TOO_FAR_FLAG 10000.0 : Value added to the FSCA quality flag to indicate that the pixel is too far off nadir;
- WATER_FLAG 1.0 : FSCA quality flag indicating that the pixel falls on water and

- #ifdef __linux__ : definition of NULL. #undef NULL NULL 0
  #endif.

4.6 Programming Style

The scag source code, with minor exceptions, adheres to the requirements set forth in the NOAA/NESDIS/STAR C programming standards and guidelines. Most of our deviations are stylistic in nature (e.g., indentation, size of subroutines).

It's appropriate to underscore a few points relating related our programming style:

1 Since the scanf function does not allow for much opportunity to check for file content errors, we felt it prudent to avoid its use. The scag program reads several ASCII files whose contents, if incorrect, will result in erroneous and unpredictable results. The content of these files are carefully scrutinized by the code.

2 While multithreaded programs are extremely efficient, they also require very deliberate coding practices to handle runtime errors. Most notably, the proper joining of thread processes prior to termination. One should not overlook the possibility of premature termination by a KILL signal. The scag program includes a critical signal handling subroutine to ensure against memory leaks and orphaned processes resulting from the interrupt (control-c) and terminate (control-z) signals. Note that sure-kill signals can not be intercepted by interrupt handlers and should therefore be avoided.

3 The FSCA algorithm was encoded using several functions. For ease of management, the code is delivered as a package of one function per file. We took liberty with the coding standards when it comes to the size of subroutines. It is our feeling that arbitrarily limiting the size of subroutines breaks the flow of concentration while reading the code.

4.7 Program Outline and Flowchart

The figures included in this section outline the organization of scag and flowchart its major elements. The following sections will describe the individual components of scag in greater detail.

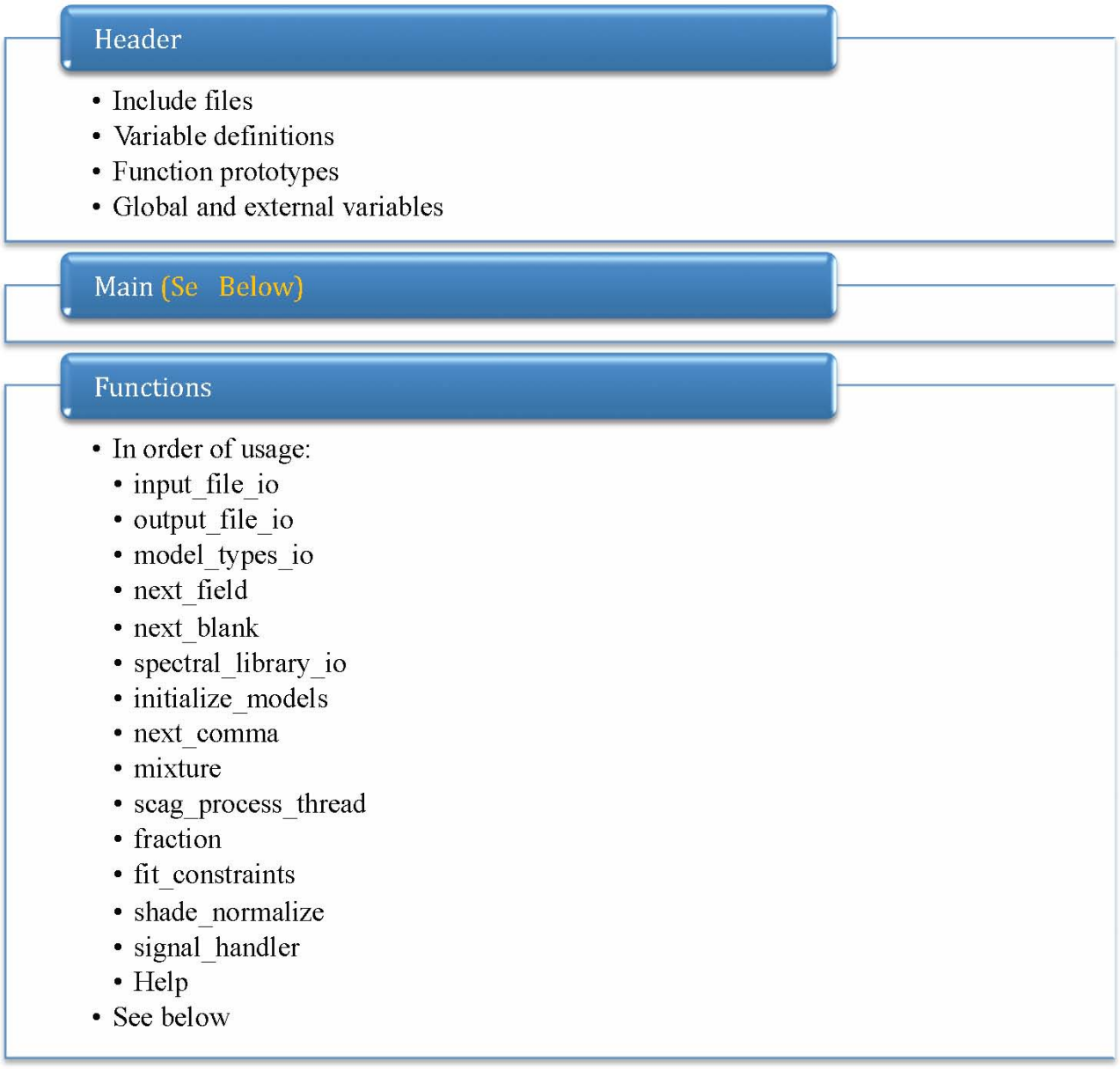
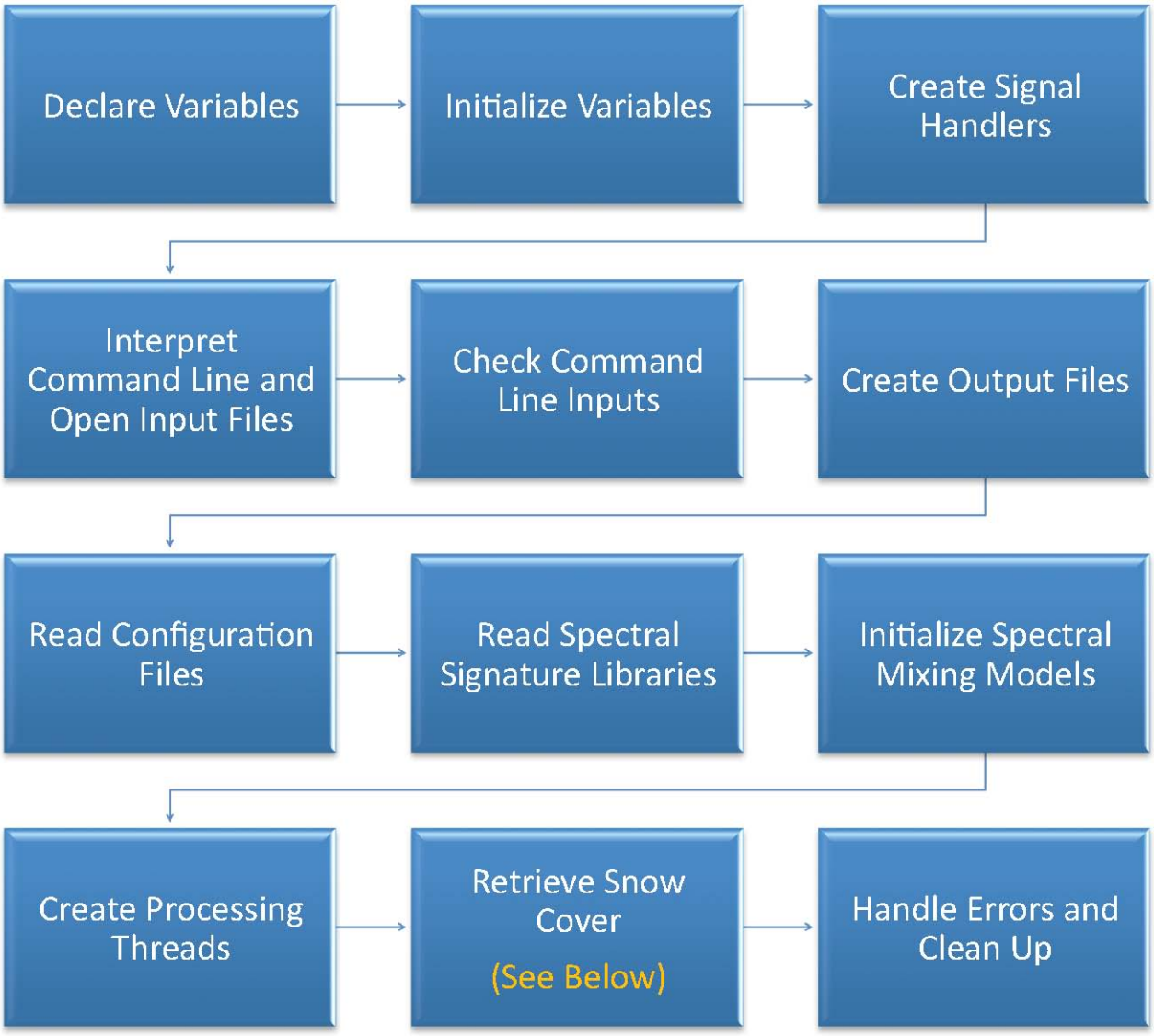


Figure 2. Basic outline of scag

Figure 9 outlines the basic structure of scag. The header includes included files, variable definition macros, function prototypes, and global and external variables. (See Appendix 1.1 for the global variables used.) The variable definition macros (see section 4.5 above) may be used to further configure (beyond the configuration files) or refine the implementation of the FSCA. Every effort was made to capture all variables, which impact the overall performance of the program in these definitions. The programmer should be confident that manipulating only these variables (and not the code) would yield the desired results (e.g., the size, resolution, and orientation of the endmember memory file).

The header is followed by the main body of the program, which handles user inputs, performs file management and I/O and orchestrates the general flow of the program (see figure 10 below).

The main body is followed by several functions, which contain the specific logic required of main's tasks and, most importantly, capture the physics and mathematics of the FSCA (see the



remaining sections of this document).

Figure 3. Flowchart of scag's main body

The flowchart for the scag's main body (main) is presented in figure 10. Briefly:

1 Declare variables: Declares variables that local to the main body;

2 Initialize Variables: Initializes both global and local variables. Initialization of global variables is critical for proper synchronization of the processing threads;

3 Create Signal Handlers: These should be created as soon as possible to ensure that critical code segments are handled as gracefully as possible in the event of unexpected termination;

4 Interpret Command Line and Open Input Files: Interprets and checks user inputs for errors then opens input files;

5 Check Command Line Inputs: Verifies that all required inputs were provided and are consistent with one another;

6 Create Output Files: All required output files are created. The output files are allowed to be over written. The endmember memory file, if it already exists, is read into memory. Otherwise it is created and initialized;

7 Read Configuration Files: Configuration files are read into memory and organized for greatest processing efficiency;

8 Read Spectral Signature Libraries: Endmember spectra are read into memory and coordinated with memory resident configuration file contents;

9 Initialize Spectral Mixing Models: Pre-calculate those portions of the model solutions that would otherwise be redundant;

10 Create Processing Threads: Create processing threads, mutexes, and thread condition variables;

11 Retrieve Snow Cover: This scag's main processing loop. It reads input data, coordinates the processing threads and writes output data. See figure 4; and

12 Handle Errors and Clean Up: At this point processing threads are files are properly terminated and joined, files closed and allocated memory freed.

The flowchart for scag's main processing loop is presented in figure 11. Up to this point each of the processing threads has been instructed to remain idle and to await further instruction from the main body of the program which current has control. The main processing loop begins by reading one row of input from each of the surface reflectance files and the accompanying rows from the static input data files. The loop proceeds by informing the process threads that they should begin the FSC retrieval model calculations for their current pixels (see figure 12). At this point in the loop, the main body of the program suspends itself (yielding control to its process threads) while waiting on the process threads as they coordinate their activities until all of the pixels on the current row have been exhausted. When the last pixel has been processed, the main body of the program receives a broadcasted thread condition signal indicating that the current row of output data is ready to be written out to disk. While the outputs are written and the next row of input is read, the individual process threads have returned to their idle states until informed to begin processing again.

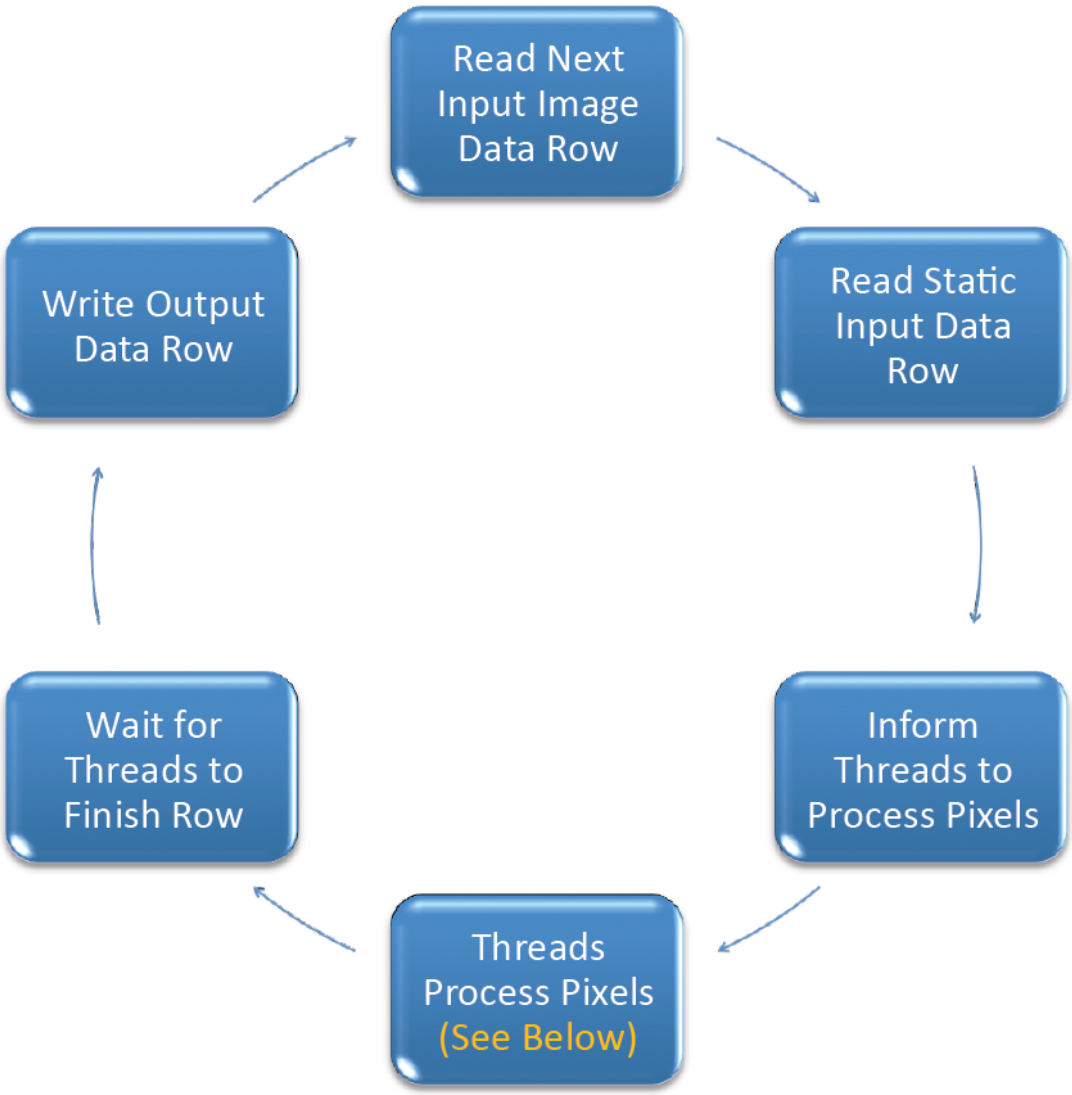


Figure 4. Flowchart of scag's main body component of FSC retrieval logic

The basic flowchart for each process thread is presented in figure 12. Each thread is essentially an infinite loop containing three basic elements:

1. The first element determines overall condition of scag's execution. If it determines that an error has occurred in another thread or in main it exits the loop. If it determines that an interrupt signal has been intercepted it exists the loop. If it determines that the last row of input has been processed it exists the loop. If it determines that the next row of input is not yet ready it continues to the top of the loop. Otherwise, the thread coordinates with all other threads to determine which pixel it's to work on next;

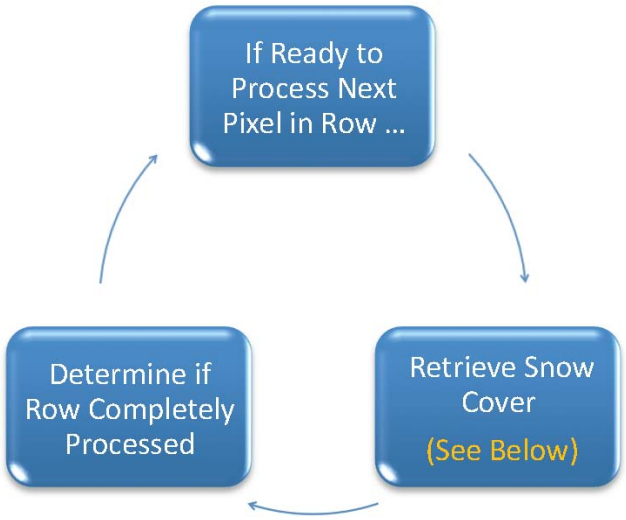


Figure 5. Basic flowchart of processing threads' FSC retrieval logic

1 The second element performs all necessary calculations required to model a FSC retrieval on the thread's current pixel (see figure 13); and

2 The last element of the loop contains the logic necessary to determine whether or not the last pixel of the current row has been successfully processed. If so, the thread broadcasts that information to main in the form of a thread condition signal. At that point the main body of the program regains control while the process threads are suspended in an idle mode.

A detailed flowchart of the second element of each thread's processing loop is presented in figure 6. This is the heart of the FSCA's implementation. It coordinates all of the FSC retrieval activities surrounding the physics and mathematics of the FSCA. The physics and mathematics are captured in subroutines that will be discussed in subsequent sections. The purpose for this section of code is to ensure that all of the proper conditions for modeling FSC retrievals for the current pixel have been met and to orchestrate the retrieval process with maximum efficiency. (One should please reacquaint oneself with section 1.1 above if necessary).

A brief description of the elements in figure 13 is as follows:

1 Skip Pixels with Missing Data: Pixels missing any of the input data and pixels with coordinates not contained within the limits of the endmember memory file are not modeled for FSC retrievals. They are represented in the output files with the no-data value and the indicated with the no data flag in the quality files;

2 Skip Water Pixels: Pixels that are associated with salt water in the land-water mask are not modeled for FSC retrievals. They are represented in the output files with the no-data value and the indicated with the water flag in the quality files;

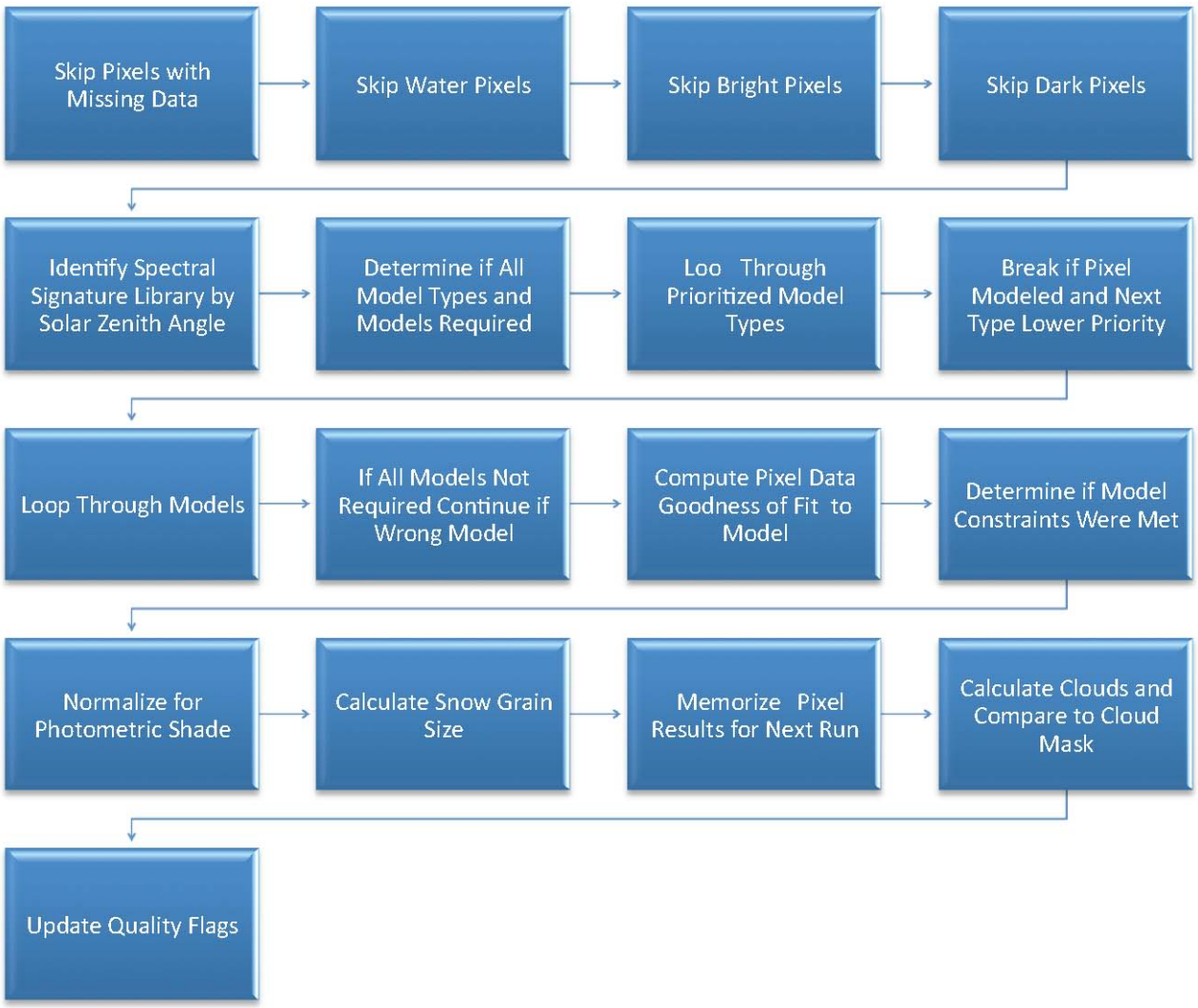


Figure 6. Detailed flowchart of processing threads' FSC retrieval logic

1 Skip Saturated Pixels: Pixels that are too saturated (low solar zenith angle) are not modeled for FSC retrievals. They are represented in the output files with the no data value and the indicated with the solar zenith angle too small flag in the quality files;

2 Skip Dark Pixels: Pixels that are too dark (high solar zenith angle) are not modeled for FSC retrievals. They are represented in the output files with the no-data value and the indicated with the solar zenith angle too large flag in the quality files;

3 Identify Spectral Signature Library by Solar Zenith Angle: Determine, based on the pixel's solar zenith angle, which of the spectral signature libraries, is the most appropriate to use. Each spectral signature library contains endmember spectra for a fixed solar zenith angle. That library with the solar zenith angle nearest the pixel's solar zenith angle is the correct one to use;

4 Determine if All Model Types and Models Required: Since the non-snow endmembers are relatively static it is not necessary to analyze each possible spectral mixing model if

knowledge of the non-snow endmembers is retained from image to image. The endmember memory file discussed in section 2.5 above retains this knowledge. Since the non-snow endmembers do gradually change it is important to periodically test each possible mixing model for a given pixel. The scag program tests each model for each pixel at its local solar noon. It also tests each model if:

a. It's never seen a non-snow endmember for the pixel;
b. The solar zenith angle has decreased or
c. The sensor zenith angle has decreased.

The later three cases should rarely occur and are probably limited to the first few executions of the program. At this point in the program a flag is set if it is not necessary to test all models;

7. Loop Through Prioritized Model Types: As discussed in section 2.3, the scag program executes several prioritized spectral mixing models types. The scag program executes these models types in descending order of priority. If a mixture model meets the constraints for a given model type priority level, there is no need to test mixture model types of lesser priority;

8. Break if Pixel Modeled and Next Type Lower Priority: See (7) above;

9. Loop through Models: Each spectral mixing model type consists of several spectral mixing models that vary by endmember composition and snow grain size. At this point in the program we loop through each possible model within the current spectral mixing model type;

10. If All Models Not Required Continue if Wrong Model: As mentioned above, for a given pixel we can skip spectral mixing models that have an endmember consistency that does not match the endmembers that were observed the last time scag ran. Item (6) above notes the possible exceptions. If it was determined in (6) above that it is not necessary to examine each model and the current model's endmember consistency does not match the last observed non-snow endmembers, the program at this point continues to the next model;
11. Compute Pixel Data Goodness of Fit to Model: At this point a solution is calculated for the current spectral mixing model. The solution consists of the best fit between the pixel's observed spectrum and the *a priori* measured endmember spectra of the model. The solution yields:

a) The fractional composition best-fit to the model's endmembers; b) The RMSE goodness of fit measure and c) The model residual for each band;

12. Determine if Model Constraints Were Met: A set of model fit constraints is associated with each of the prioritized spectral mixing model types. A pixel is considered modeled only if these model constraints have been met. Of course several models within a model type [assuming all models have to be examined – see (6) and (10) above] may meet these constraints for a given pixel. That model with the smallest RMSE is the one that is retained for the pixel. If no model meets the constraints for a given pixel, that pixel is flagged as such in the quality file. Remember, if the constraints are met for a spectral mixing model within the current model type, the spectral mixing models in model types of lower priority are not evaluated;

13. Normalize for Photometric Shade: Each spectral mixing model actually accounts for an additional end member – photometric shade. The calculated fraction of shade is proportionally reallocated among the non-shade endmembers;

14. Calculate Snow Grain Size: There are 110 snow endmembers in each spectral mixing model type. Each of the 110 snow endmembers is associated with a snow grain size. At this point a grain size is associated with the snow endmember if one exists;

15. "Remember" Pixel Results for Next Run: Record the endmember composition, solar zenith angle and sensor zenith angle for the current pixel in the endmember mask. This information will be used during the next scag execution as described in (6) and (10) above;

16. Calculate Clouds and Compare to Cloud Mask: If snow if determined to exist on the current pixel and its grain size meets or falls below a specified radius, it is considered be cloud and indicated as such in the quality file. Cloud based on snow grain size is compared against the ACM and various cloud comparison qualifiers are indicated in the quality file; and

17. Update Quality Flags: The quality flag is updated to indicate whether or not the pixel is close enough to nadir or too dim to be properly modeled.

4.8 Functions

4.8.1 Overview

Each function is contained within its own file, the name of which closely corresponds with the function within (e.g., process_thread.c contains the function process_thread). Some of the functions have corresponding error functions, as appropriate. Each of these functions are also in their respective files (e.g., error_process_thread.c contains the function error_process_thread).

4.8.2 main

The main function is the driving function for the algorithm. While the below outline appears short, it is deceptively so, as much of the processing is done by the main functions.

Here is a brief outline of the main functional blocks of the main program:

1 Define and initialize variables
2 Define signal handlers
3 Parse command line and open input files
4 Create output files
5 Read configuration files and spectral libraries
6 Initialize thread data and create process threads
7 Calculate fractional snow cover and grain size

This function has a related error function, error_main, which contains code for cleaning up in the event of an error in main.

See Appendix 1.2 for code segments.

4.8.3 input_file_io

This function reads into memory the contents of the source data and ancillary data. This function has a related error function, error_input_file_io, which contains code for cleaning up in the event of an error in input_file_io. The code for these functions is not relevant to this document.

4.8.4 output_file_io

This function writes to disk files of processed results. This function has a related error function, error_output_file_io, which contains code for cleaning up in the event of an error in output_file_io. The code for these functions is not relevant to this document.

4.8.5 model_types_io

The model_types_io function reads spectral unmixing model configuration file data into memory. These data are held in memory for the duration of the program. The configuration file data are organized with the spectral library data by the initialize_models function (see section
4.8.9 below). The model_types_io function has three modes of operation. Their purpose and a brief outline of their structure are as follows:

1. Terminate option – Systematically free all memory allocated for the spectral unmixing model configuration data; and
2. Read spectral unmixing model configuration file data:
a. Check arguments;
b. Read spectral unmixing model types configuration file data;
c. Sort spectral unmixing model types into descending order of priority. This will improve the efficiency of the process_thread function (see section 4.8.10 below);
d. Read the model endmember types into memory;
e. Read the snow grain size lookup table into memory and
f. Read the model endmembers into memory.

This function has a related error function, error_model_types_iuo, which contains code for cleaning up in the event of an error in model_type_io, including a terminate option.

See Appendix 1.3 for code segments.

4.8.6 next_field

This function finds the next field in a string. Its code is not relevant to this document.

4.8.7 next_blank

This function finds the next blank character in a string. Its code is not relevant to this document.

4.8.8 spectral_library_io

The spectral_library_io function reads the spectral library configuration file, allocates memory to retain the spectral library contents for the duration of the scag program, reads the spectral libraries into memory and calculates the best spectral library to use for a given pixel's solar zenith angle. The spectral_library_io function has three modes of operation. Their purpose and a brief outline of their structure are as follows:

1. Terminate option – Systematically free all memory allocated for the spectral libraries;
2. Read spectral libraries into memory option – Read the spectral library configuration file and load the spectral libraries into memory:
a. Check the arguments;
b. Loop through the configuration file records. There is one record for each spectral library;
i. Parse the configuration file record variables and retain in memory and
ii. Read the next spectral library and retain in memory and
3. Calculate best spectral library option – Based on a pixel's solar zenith angle, calculate the most appropriate spectral library to use in the spectral unmixing calculations:

a. Loop through the spectral library solar zenith angles;
    i. `Calculate the absolute value of the difference between the pixel's solar zenith angle and the next spectral library's solar zenith angle and
    ii. If the pixel's angle is closer to this library's angle than previous libraries' angles retain the distance and the library's index (for use in spectral unmixing calculations).

This function has a related error function, error_spectral_types_io, which contains code for cleaning up in the event of an error in model_spectral_io, including a terminate option.

See Appendix 1.4 for code segments.

4.8.9 initialize_models

The initialize_models function reads the model constraints configuration file into memory, allocates spectral unmixing model memory, and organizes configuration file and spectral library file data in preparation for FSC retrievals. These data are held in memory for the duration of the program. Lastly, it calls the mixture function (see section 4.8.11 below) for a one time only initialization of matrices for all of the unmixing models. The initialize_model function has two modes of operation. Their purpose and a brief outline of their structure are

as follows:

1. Terminate option – Systematically free all memory allocated for the spectral unmixing models; and
2. Initialize spectral unmixing models – Read the model constraint configuration file into memory, allocate spectral unmixing model memory, organize configuration file and spectral library data for models, and initialize the models:
a. Check the arguments;
b. Read the model constraints file into memory;
c. Relate the spectral unmixing model types with their associated constraints;
d. Allocate memory for each spectral unmixing model;
e. Organize configuration file and spectral library file data into spectral unmixing model memory and
f. Initialize each spectral unmixing model using the mixture function (see section

4.8.9 below).

This function has a related error function, error_initialize_models_io, which contains code for cleaning up in the event of an error in model_initialize_models_io, including a terminate option.

See Appendix 1.5 for code segments.

4.8.10next_comma

This function finds the next comma in a string. Its code is not relevant to this document.

4.8.11 mixture

The mixture function initializes the matrices used to calculate the endmember fractions and goodness of fit metrics [RMSE and per band model residuals used by the fit_constraints function (see 4.8.12 below)] yielding the best fit between the spectrum of sensor measured surface reflectance for a pixel and the *a priori* measured pure endmember spectra of a spectral unmixing model. The matrices for the unmixing models are used in the fraction function (see 4.8.11 below). The mathematical basis and details are presented in section 1.1 above.

See Appendix 1.6 for code segment.

4.8.12 process_thread

Refer to the detailed description accompanying figure 12 in section 4.7 above for an outline of the process_thread function.

This function has a related error function, error_process_thread_io, which contains code for cleaning up in the event of an error in model_process_thread_io.

See Appendix 1.7 for code segments.

4.8.13 fraction

The fraction function calculates the endmember fractions and goodness of fit metrics [RMSE and per band model residuals used by the fit_constraints function (see 4.8.14 below)] yielding the best fit between the spectrum of sensor measured surface reflectance for a pixel and the *a priori* measured pure endmember spectra of a spectral unmixing model. The matrices for the unmixing models are initialized in the mixture function (see 4.8.11 above). The mathematical basis and details are presented in section 1.1 above.

See Appendix 1.8 for code segment.

4.8.14 fit_constraints

The fit_constraints function determines whether or not the results of the spectral unmixing model fit the constraints associated with its mixing model type. There are three tests:

1 Check to see if the model's RMSE exceed the model type's threshold RMSE;
2 Check to see if the modeled endmember fractions exceed the model type's minimum or maximum threshold fractions; and
3 Check to see if the model residuals for N consecutive input surface reflectance bands exceed the model type's threshold residual (where N is a configuration variable).

See Appendix 1.9 for code segment.

4.8.15 shade_normalize

Each of the spectral mixing model types (one- and two-endmember model types) actually model and additional endmember – photometric shade. The modeled shade fraction is proportionally redistributed among the non-shade endmembers. The math is straightforward and requires no further description.

See Appendix 1.10 for code segment.

4.8.16 signal_handler

The signal_handler function handles the intercepted SIGINT and SIGTERM signals. While not critical for memory or file handling, it is extremely critical for process thread handling. Unless process threads are correctly terminated and joined, there is the potential for memory leaks and orphaned processes.

See Appendix 1.11 for code segment.

4.8.17 help

This function displays usage and other important information for the successful execution of the program. Its code is not relevant to this document.

5 REFERENCES

Keshava, N (2003). A Survey of Spectral Unmixing Algorithms. Lincoln Laboratory Journal, Volume 14, Number 1, 55 – 78.

APPENDIX 4: GOES-R SURFACE REFLECTANCE ALGORITHM THEORETICAL BASIS DOCUMENT

GOES-R Advanced Baseline Imager (ABI) Algorithm Theoretical Basis For Surface Albedo and Surface Reflectance

Shunlin Liang, Dongdong Wang, Tao He

University of Maryland

Yunyue Yu, NOAA/NESDIS/STAR

LIST OF ACRONYMS

2D	Two Dimension
ABI	Advanced Baseline Imager
ACM	ABI Cloud Mask
AIT	Algorithm Integration Team
AOD	Aerosol Optical Depth
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
ATB	Algorithm Theoretical Basis
AVHRR	Advanced Very High-Resolution Radiometer
BRDF	Bidirectional Reflectance Distribution Function
BRF	Bidirectional Reflectance Factor
ETM+	Enhanced Thematic Mapper Plus
FD	Full Disk
GOES	Geostationary Operational Environmental Satellite
GS-F&PS	Ground Segment Functional and Performance Specification
L1B	Level 1B
LSA	Land Surface Albedo
LUT	Look Up Table
MFRSR	Multi-Filter Rotating Shadowband Radiometers
MISR	Multi-angle Imaging Spectroradiometer
MODIS	Moderate Resolution Imaging Spectroradiometer
MRD	Mission Requirement Document
MSG	Meteosat Second Generation

NCEP	National center for Environmental Prediction
NEDT	Noise Equivalent Delta Temperature
NESDIS	National Environmental Satellite, Data, and Information Service
NOAA	National Oceanic and Atmospheric Administration
PAR	Photosynthetically Active Radiation
POLDER	Polarization and Directionality of the Earth's Reflectance
PQI	Product Quality Information
PSP	Precision Spectral Pyranometer
QF	Quality Flag
QC	Quality Control
SEVIRI	Spanning Enhanced Visible and Infrared Imager
SNR	Signal Noise Ratio
SPOT	Systeme pour l'Observation de la Terre
STAR	Center for Satellite Applications and Research
SURFRAD	SURFace RADiation network
TOA	Top Of Atmosphere
UTC	Coordinated Universal Time
VIIRS	Visible/Infrared Imager /Radiometer Suite

1 INTRODUCTION

The purpose, users, scope, related documents and revision history of this appendix are briefly described in this section. Section 2 gives an overview of the Advanced Baseline Imager (ABI) Land Surface Albedo (LSA) algorithm derivation objectives and operation concept. Section 3 describes the LSA algorithm, its input data requirements, the theoretical background, mathematical descriptions and output of the algorithm. Some test results will be presented in Section 4. Practical considerations are described in Section 5, and followed by Section 6 on assumptions and limitations. Finally, Section 7 presents the references cited.

1.1 Purpose of This Appendix

The LSA Algorithm Theoretical Basis (ATB) provides a high level description and the physical basis for the estimation of land surface albedo and land surface reflectance with images taken by ABI onboard the Geostationary Environmental Operational Satellite (GOES) R series of NOAA geostationary meteorological satellites. The LSA is a key parameter controlling surface radiation and energy budgets. LSA and land surface reflectance are also needed by other algorithms, such as snow coverage and radiation flux products.

1.2 Who Should Use This Appendix

The intended users of this appendix are those interested in understanding the physical basis of the algorithms and how to use the output of this algorithm to optimize the albedo and surface reflectance estimate for a particular application. This appendix also provides information useful to anyone maintaining or modifying the original algorithm.

1.3 Inside Each Section

This appendix is subdivided into the following main sections:

- **System Overview**: Provides relevant details of the ABI and a brief description of the products generated by the algorithm.

- **Algorithm Description**: Provides a detailed description of the algorithm including its physical basis, its input, and its output.

- **Test Data Sets and Output**: Provides a description of the test data sets for characterizing the performance of the algorithm and quality of the data products. It also describes the results from algorithm processing using simulated input data.

- **Practical Considerations**: Provides an overview of the issues involving in numerical computation, programming and procedures, quality assessment and diagnostics and exception handling.

- **Assumptions and Limitations**: Provides an overview of the current limitations of the approach and gives the plan for overcoming these limitations with further algorithm development.

1.4 BRF and BRDF

Bidirectional reflectance factor (BRF) and bidirectional reflectance distribution function (BRDF) are two concepts that will be used frequently in this appendix. BRF is one kind of reflectance, a ratio between outgoing radiance at one given direction and incoming radiance at another given direction (same or different from the incoming direction). The reflectance byproduct of the LSA algorithm is the product of BRF. BRDF is a model to describe the bi-directional properties of reflectivity. In this appendix, BRDF is also used in the term "BRDF parameters" to refer to the kernel coefficients of the BRDF model.

1.5 Related Documents

LSA is one product of ABI product streamlines. The requirements of LSA products can be found in the specifications of the GOES-R Ground Segment Functional and Performance Specification (F&PS). LSA also requires other ABI products as the algorithm input. The readers can refer to these specific ATBDs for more information:

- *GOES-R Algorithm Theoretical Base Document for ABI Aerosol Optical Depth*
- *GOES-R Algorithm Theoretical Base Document for ABI Cloud Mask*

More references about the algorithm details are given in Section 5.

1.6 Revision History

This appendix was created by Drs. Shunlin Liang and Kaicun Wang of the Department of Geography, University of Maryland, College Park and Dr. Yunyue Yu of NOAA NESDIS, Center for Satellite Applications and Research, Camp Springs, Maryland. Refinement of the algorithm and latest validation results were updated by Drs. Shunlin Liang, Dongdong Wang and Mr. Tao He of the Department of Geography, University of Maryland, College Park, and Dr. Yunyue Yu of NOAA.

2 OBSERVING SYSTEM OVERVIEW

This section describes the products generated by the ABI LSA algorithm and the requirements it places on the sensor.

2.1 Products Generated

This albedo algorithm is responsible for estimation of LSA and land surface BRF for clear sky pixels identified by the ABI Cloud Mask (ACM) product. Using the ABI Aerosol Optical Depth (AOD) product as the first guess, this algorithm updates AOD and estimates AOD at points where ABI AOD products are not available, and then retrieves the parameters of the land surface BRDF model and derive LSA and land surface BRF values. It also incorporates albedo climatology from previous satellite products (MODIS) as prior knowledge. Full disk blue-sky broadband albedo for the solar zenith angle smaller than 67° is produced. As a byproduct, full disk surface BRFs at the five bands are generated as well.

The surface albedo/reflectance product requirements defined by the Mission Requirement Document (MRD) and the Ground Segment Functional and Performance Specification (GS-F&PS) (NOAA 2009) are listed in Tables 2.1 and 2.2.

Table 2.1. GOES-R mission requirements for surface albedo product

Observational Requirement	Geographic Coverage[2]	Horiz. Res.	Mapping Accuracy	Msmnt. Range (albedo unit)	Msmnt. Accuracy (albedo unit)	Msmnt. Precision	Refresh Rate	Data Latency	Long-term Stability	Exent Qualifier
Albedo: Full Disk	FD	2 km	2 km	0 to 1	0.08	10%	60 mins	3236 secs	TBD	LZA <70

Table 2.2. Requirements for surface reflectance product

Observational Requirement	Geographic Coverage[2]	Horiz. Res.	Mapping Accuracy	Msmnt. Range	Msmnt. cAccuray	Msmnt. Precision	Refresh Rate	Data Latency	Long-term Stability	Extent Qualifier
Reflectance: Full Disk	FD	2 km	2 km	0 to 2	0.08	0.08	60 mins	3236 secs	TBD	LZA <70

As the key component of surface energy budget, satellite albedo products can be used to drive/calibrate/validate climatic, mesoscale atmospheric, hydrological and land surface models. Variation of LSA is also an important indicator of land cover and land use change. Analysis of long-term reliable albedo products will help better understand the human dimension of climate change and how the vegetation-albedo-climate feedbacks work. The land surface reflectance

byproducts will be the input to a number of other high-level land surface products, such as the factional snow cover product.

2.2 Instrument Characteristics

The LSA product is produced from clear-sky pixels observed by the ABI. The final channel set is still being determined as the algorithms are developed and validated. Table 2.3 highlights the ABI channels used by the albedo algorithm.

Table 2.3.Spectral characteristics of Advanced Baseline Imager

Channel Number	Central Wavelength (μm)	Bandwidth (μm)	Spatial Resolution
1	0.47	0.45 – 0.49	1 km
2	0.64	0.59 – 0.69	0.5 km
3	0.86	0.85– 0.89	1 km
4	1.38	1.37 – 1.39	2 km
5	1.61	1.58 – 1.64	1 km
6	2.26	2.23 – 2.28	2 km
7	3.9	3.80 – 4.00	2 km
8	6.15	5.77 – 6.60	2 km
9	7.0	6.75 – 7.15	2 km
10	7.4	7.24 – 7.44	2 km
11	8.5	8.30 – 8.70	2 km
12	9.7	9.42 – 9.80	2 km
13	10.35	10.10 – 10.60	2 km
14	11.2	10.80 – 11.60	2 km
15	12.3	11.80 – 12.80	2 km
16	13.3	13.00 – 13.60	2 km

Shaded channels are used for Albedo production.

3 ALGORITHM DESCRIPTION

This section provides a complete description of the algorithm, including both theoretical basis and technical details.

3.1 Algorithm Overview

Three steps are typically required to estimate the surface albedo from satellite multispectral TOA observations (Liang 2004; Schaaf et al. 2008):

(1) atmospheric correction,
(2) surface directional reflectance modeling,
(3) narrowband-to-broadband conversion.

The typical example is the MODIS surface albedo algorithm. The first step converts TOA reflectance into surface reflectance, the second step converts surface spectral reflectance into spectral albedos (individual ABI bands), and the last step converts spectral albedos to a broadband albedo. Instead, we propose an optimization method similar to the earlier algorithm used on the Meteosat data (Pinty et al. 2000a, b) to directly retrieve surface BRDF parameters, and then use the derived BRDF parameters to calculate LSA and land surface BRF. A similar strategy was also used to retrieve daily aerosol and surface reflectance simultaneously from the Spinning Enhanced Visible and Infrared Imager (SEVIRI) on the Meteosat Second Generation (MSG) (Govaerts et al. 2010; Wagner et al. 2010). Our proposed algorithm combines atmospheric correction and surface BRDF modeling together in one optimizing code. The optimization process estimates the BRDF parameters by minimizing a cost function considering both TOA reflectance and albedo climatology. Our revision over the previous methods mainly includes:

- AOD can vary over time;
- we use multiple ABI spectral channels enabling accurate production of shortwave broadband albedo;
- we use a different formulation of the atmospheric radiative transfer and surface BRDF model; and
- we incorporate albedo climatology as the constraint of optimization.

Both our optimization algorithm and traditional MODIS algorithm need aggregation of multiple clear-sky observations within a relatively short time periods, during which the surface conditions keep relatively stable. Continuous cloud coverage and rapidly changing surface usually cause the failure of the routine LSA algorithm. To handle these cases, we use the direct estimation approach as the back-up algorithm.

3.2 Processing Outline

The retrieval of BRDF parameters needs multiple observations over varied observing geometries. Since ABI is not a multi-angular sensor, we achieve this by using a stack of time series observations over each pixel within a short period time and assume the BRDF parameters are relatively stable during the compositing period (7 days). The organization of time series data and

retrieval of BRDF parameters are a time-consuming process. In order to improve the code efficiency, we divide our algorithm into two parts: the offline mode and the online mode. At the end of each day, an offline mode computation is conducted to perform a full inversion of BRDF parameters using the stacked time series data at 15-mins intervals. The calculated BRDF parameters are saved for the usage of the online mode next day. When the abovementioned routine algorithm fails, the back-up algorithm called direct estimation approach will be invoked to calculate broadband albedo directly in the online mode. Otherwise, the pre-calculated BRDF parameters are used to derive full disk LSA products every 60 minutes in the online mode. In addition to albedo, surface BRF will also be generated every 60 minutes in the online mode using AOD and/or the retrieved BRDF model parameters from the offline mode. Although ABI has a refreshing rate of 15 mins, surface albedo products are required to be generated every 60 mins. The online mode will select the most clear and latest observation among the four observations within the 60 mins window to calculate LSA and BRF. The processing chains of the LSA algorithm offline and online modes are shown in the Figures 3.1 and 3.2, respectively.

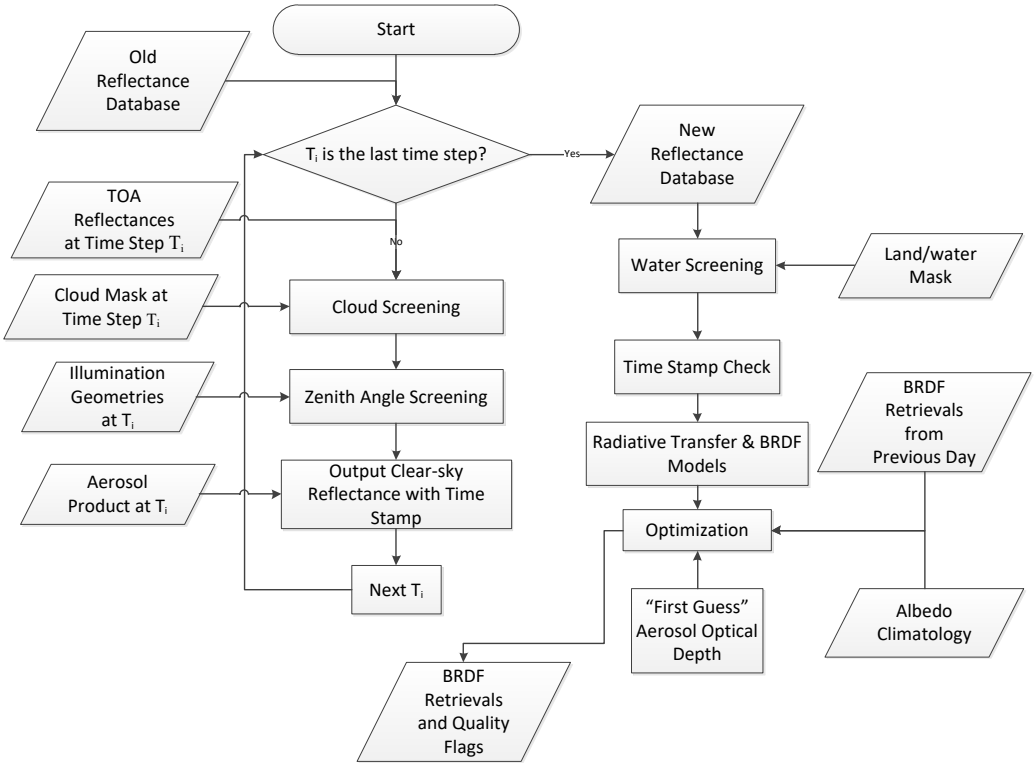


Figure 3.1. High level flowchart of the offline mode of ABI LSA algorithm, which is executed once at the end of each day to estimate the BRDF parameters.

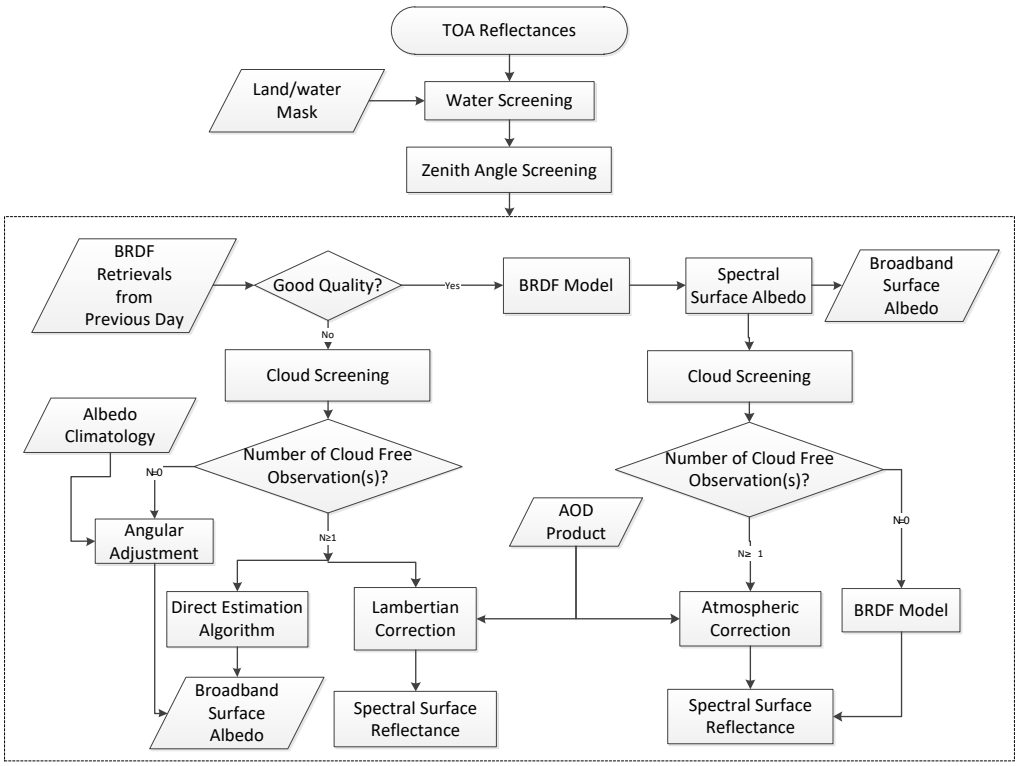


Figure 3.2. High level flowchart of the online mode of ABI LSA algorithm, illustrating the main processing components.

The LSA algorithm will take the ABI AOD product as one input, however, the accuracy and integrity of the AOD product may need to be improved for accurate LSA estimation. For bright surface types or other conditions where the ABI AOD products are not available, the AOD information will be solely obtained from the LSA algorithm. Thus, the ABI AOD products are used as the "first-guess" values. Our strategy is to estimate the land surface BRDF parameters and update AOD information simultaneously based on the initial estimates of AOD and albedo climatologies.

3.3 Algorithm Input

This section describes the input required to execute the LSA algorithm. The offline mode and online mode have different requirements. Tables 3.1 and 3.2 list them respectively.

Table 3.1. Summary of inputs for ABI LSA algorithm offline mode.

Sensor input (One day's data)		TOA reflectance at five bands
		Viewing zenith angle
		Solar zenith angle
		Relative azimuth angle
		Geolocation
Ancillary data	**ABI data (One day's data)**	Aerosol optical depth and aerosol type
		Cloud mask

	Non-ABI static data	Surface albedo climatology
		LUT
		Land/ocean mask
	Non-ABI dynamic data	Clear-sky Observation Database
		BRDF and its QC of previous day
		Configuration file

Table 3.2. Summary of inputs for ABI LSA algorithm online mode.

Sensor input (60 minute's data)		TOA reflectance at five bands
		Viewing zenith angle
		Solar zenith angle
		Relative azimuth angle
		Geolocation
Ancillary data	**ABI data (60 minute's data)**	Aerosol optical depth
		Cloud mask
	Non-ABI static data	Surface albedo climatology
		LUT
		Coefficients of direct estimation approach
		Land/ocean mask
	Non-ABI dynamic data	Pre-calcualted BRDF parameters
		Configuration file

Basically, the offline mode algorithm needs the time series of all types of input data and the online mode involves only the data sets up to 60 minutes ahead of the current observation time since the online mode algorithm is only run once an hour. Moreover, the online mode needs the pre-calculated BRDF parameters as the input. For one particular data set, the online mode and offline mode share the same details, which are given in the following subsections.

3.3.1 Primary Sensor Data

Primary sensor data is information that is derived solely from the ABI observations. The primary sensor data used by the LSA algorithm include both the TOA radiance values and relevant ancillary information (angles and geolocation). The spatial resolution of LSA and BRF products are 2km, the input ABI sensor data should be aggregated to 2km as well.

Table 3.3. Input list of primary sensor data.

Name	Type	Description	Dimension
Ch1	input	Calibrated ABI level 1b reflectance at channel 1	grid (xsize, ysize)

Ch2	input	Calibrated ABI level 1b reflectance at channel 2	grid (xsize, ysize)
Ch3	input	Calibrated ABI level 1b reflectance at channel 3	grid (xsize, ysize)
Ch5	input	Calibrated ABI level 1b reflectance at channel 5	grid (xsize, ysize)
Ch6	input	Calibrated ABI level 1b reflectance at channel 6	grid (xsize, ysize)
Latitude	input	Pixel latitude	grid (xsize, ysize)
Longitude	input	Pixel longitude	grid (xsize, ysize)
Solar zenith	input	ABI solar zenith angles	grid (xsize, ysize)
Solar azimuth	input	ABI solar azimuth angles	grid (xsize, ysize)
View zenith	input	ABI view zenith angle	grid (xsize, ysize)
View azimuth	input	ABI view azimuth angle	grid (xsize, ysize)

3.3.2 Derived Sensor Data

There are two ABI derived sensor data products required by the LSA algorithm: 1) the ABI Cloud Mask (ACM) product, which indicates four cloudiness states for each pixel: clear, probably clear, probably cloudy, and cloudy, and 2) the ABI AOD.

Table 3.4. Input list of derived sensor data.

Name	Type	Description	Dimension
Cloud mask	input	ABI cloud mask product	grid (xsize, ysize)
AOD	input	ABI AOD product	grid (xsize, ysize)

3.3.3 Ancillary Data

Ancillary data are data other than the ABI sensor and derived data (Table 3.5). The following lists and briefly describes the ancillary data required to run the LSA algorithm.

- **Land/water mask**
  The ABI LSA products are generated over land pixels only. The ABI standard static Land/water mask is used to mask water pixel.

- **LSA Climatology**
  The albedo climatology includes the mean and variance of land surface spectral and broadband albedos. The albedo climatology will be used as the background values in the albedo estimation. Multiple years' MODIS albedo products are averaged and used as climatology. The MODIS albedo climatology is calculated at 8-day intervals.

- **Look-Up table**

  In order to improve the computational efficiency, the atmospheric parameters have been pre-calculated using the 6S simulation and stored into the look-up table (LUT). LUT is a type of static input to the algorithm and all codes share the same set of LUT. The parameters in the LUT include:

- Atmospheric intrinsic reflectance
- Total global gas transmittance
- Downward total scattering transmittance
- Upward total scattering transmittance
- Total spherical albedo
- Optical depth
- Direct irradiance ratio

  After a sensitivity analysis, we select the entries of our LUT by balancing the accuracy and the computational efficiency (Table 3.6).

- **BRDF model parameters**

  BRDF model parameters are useful for integrating albedo and correcting surface reflectance. The parameters are the output of the offline mode code and also the input of the online mode code.

- **Coefficients of direct estimation approach**

  Two groups of coefficients to calculate black-sky and white-sky albedo respectively from TOA spectral reflectance are stored for each observing geometry. The intervals for viewing geometries are the same as used in the LUT (Table 3.6).

Table 3.5 Input of ancillary data.

Name	Type	Description	Dimension

Land/water mask	input	A land-water mask	grid (xsize, ysize)
Albedo climatology	input	MODIS multiple years' mean	grid (xsize, ysize)
Atmosphere LUT	input	Seven atmospheric parameters as function of aerosol model, aerosol optical depth, ABI channel and observing geometry	(17x 17 x 19 x 10 x 5)*
Coefficient of direct estimation approach	input	Two groups of coefficients for calculating black-sky and white-sky albedo respectively for each observing geometry	(17x 17 x 19 x 6 x 2)
Ch1 f_iso	input	BRDF isotropic component parameter at Ch1	grid (xsize, ysize)
Ch1 f_vol	input	BRDF volumetric kernel parameter at Ch1	grid (xsize, ysize)
Ch1 f_geo	input	BRDF geometric kernel parameter at Ch1	grid (xsize, ysize)
Ch2 f_iso	input	BRDF isotropic component parameter at Ch2	grid (xsize, ysize)
Ch2 f_vol	input	BRDF volumetric kernel parameter at Ch2	grid (xsize, ysize)
Ch2 f_geo	input	BRDF geometric kernel parameter at Ch2	grid (xsize, ysize)
Ch3 f_iso	input	BRDF isotropic component parameter at Ch3	grid (xsize, ysize)
Ch3 f_vol	input	BRDF volumetric kernel parameter at Ch3	grid (xsize, ysize)
Ch3 f_geo	input	BRDF geometric kernel parameter at Ch3	grid (xsize, ysize)
Ch5 f_iso	input	BRDF isotropic component parameter at Ch5	grid (xsize, ysize)
Ch5 f_vol	input	BRDF volumetric kernel parameter at Ch5	grid (xsize, ysize)
Ch5 f_geo	input	BRDF geometric kernel parameter at Ch5	grid (xsize, ysize)
Ch6 f_iso	input	BRDF isotropic component parameter at Ch6	grid (xsize, ysize)
Ch6 f_vol	input	BRDF volumetric kernel parameter at Ch6	grid (xsize, ysize)
Ch6 f_geo	input	BRDF geometric kernel parameter at Ch6	grid (xsize, ysize)

* num_solar_zenith_angle * num_sensor_senith_angle * num_relative_azimuth_angle * num_aerosol_optical_depth * num_bands

Table 3.6 Entries of LUT.

Entries to LUT	Values
Solar Zenith Angle	0.,5.,10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70.,75., 80.
Sensor Zenith Angle	0.,5.,10.,15.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70.,75., 80.
Relative Azimuth Angle	0.,10.,20.,30.,40.,50.,60.,70.,80.,90.,100.,110., 120.,130.,140.,150.,160.,170.,180.
Aerosol Optical Depth	.01,.05,.1,.15,.2,.3,.4,.6,.8,1.

- **Clear-sky observation database**

  The database stores the clear-sky observations for each time step during a day (96 in total). The clear-sky observation for each time step is collected from the available day closest to the current day. The time stamp and solar angular information will be stored as well into the database. This database will be updated each day by absorbing the newest clear-sky observations with the time stamp and solar angular information (Table 3.7).

  The database includes both inputs and outputs of the offline mode calculations. At the end of each day, the offline code examines all the new observations and updates the database. This maintenance procedure of the offline mode mainly includes the following steps:

1) read in the reflectance database and set the time step T_i to the beginning of the day (T_0);
2) read in the TOA reflectance data the corresponding cloud mask at time step T_i;
3) check the observing geometries: if the sensor zenith is larger than 70° or the solar zenith is larger than 67°, the old data in the database will be kept and continue with the next time step from 2); otherwise continue with 4);
4) check if the pixel is cloud contaminated (cloud present or cloud shadow present) using the cloud mask and other constraints on spectral reflectances: if yes, the old data in the database will be kept and continue with the next time step from 2); otherwise continue with 5);
5) update the database with current TOA reflectance, together with its corresponding acquisition time, geometries and AOD product;
6) go back to 2) until the end of the day

Table 3.7 Details of Clear-sky observation database

Name	Type	Description	Dimension
Ch1	input	Calibrated ABI level 1b reflectance at Ch1	grid (xsize, ysize, T_i)
Ch2	input	Calibrated ABI level 1b	grid (xsize,

		reflectance at Ch2	ysize, T_i)
Ch3	input	Calibrated ABI level 1b reflectance at Ch3	grid (xsize, ysize, T_i)
Ch5	input	Calibrated ABI level 1b reflectance at Ch5	grid (xsize, ysize, T_i)
Ch6	input	Calibrated ABI level 1b reflectance at Ch6	grid (xsize, ysize, T_i)
Latitude	input	Pixel latitude	grid (xsize, ysize)
Longitude	input	Pixel longitude	grid (xsize, ysize)
Solar zenith	input	ABI solar zenith angles	grid (xsize, ysize, T_i)
Relative azimuth	input	ABI relative azimuth angles	grid (xsize, ysize, T_i)
View zenith	input	ABI view zenith angle	grid (xsize, ysize)
Time stamp	input	Julian day of the clear-sky observation at T_i	grid (xsize, ysize, T_i)

3.4 Theoretical Description

After analyzing existing albedo algorithms, we proposed an ABI LSA algorithm similar to the earlier algorithm used on Meteosat data (Pinty et al. 2000a, b) and the approach tested on MSG/SEVIRI data (Govaerts et al. 2010; Wagner et al. 2010). The albedo algorithm from the geostationary Meteosat observations combined atmospheric correction and BRDF modeling by assuming one unknown constant AOD for the whole period of time (daily). Here, we made several major revisions. For example, AOD can vary over time, and multiple ABI spectral bands enable production of multiple broadband albedos. In addition, the formulation of the atmospheric radiative transfer and surface BRDF model are also different.

This optimization approach will serve as the routine algorithm to derive BRDF parameters in the offline mode (Section 3.4.1). The next section describes the mathematical foundation and procedure used in the online mode.

3.4.1 The offline mode

The critical step in retrieving LSA and land surface BRF is to estimate the surface BRDF parameters. This procedure is time-consuming and carried out once each day in the offline mode. The tasks of the offline mode mainly include: 1) maintenance of clear-sky TOA reflectance database, and 2) retrieval of BRDF parameters using this database. The first task deals mainly with data I/O, and the programing procedure has been briefly introduced in Section 3.3.3 and will not be discussed here. The second part is the core of the LSA routine algorithm and will be introduced in details here. The mathematical

equations used in the retrieval of BRDF parameters are given in Section 3.4.1.1. Section 3.4.1.2 discusses how to optimize and derive spectral BRDF parameters from TOA database using these equations.

3.4.1.1 Mathematical formulation

In order to obtain BRDF parameters, we need to execute the atmospheric correction and BRDF modeling. A traditional way (e.g. the MODIS albedo algorithm) to achieve this is to implement them separately in two steps. Here, we achieve this in one step by combining both the atmospheric radiative transfer process and BRDF modeling in our optimization schema. Three groups of equations are introduced here: the BRDF model, the Atmospheric radiative transfer equations, and equations calculating albedo from BRDF parameters.

3.4.1.1.1 Land surface BRDF model

Performance of albedo retrieval from satellite observations is usually restricted by a limited sampling of directional surface reflectance. Therefore, a model is usually used to characterize the surface anisotropy. The model can be inverted with a finite set of angular samples and used to calculate surface reflectance in any sun-view geometry and to derive surface albedo. An empirical kernel-based BRDF model will be used in the ABI LSA algorithm.

Maignan et al. (2004) found that among the current BRDF models, the best two are the three-parameter linear Ross–Li model and the nonlinear Rahman–Pinty–Verstraete model. However, all models fail to accurately reproduce the sharp reflectance increase close to the backscattering (hotspot peak) direction. Based on physical considerations, Maignan et al. (2004) suggested a modification of the Ross–Li model, without the addition of a free parameter, to account for the complex radiative transfer within the land surfaces that leads to the hot spot signature. They illustrated that the modified linear model performs better than all others.

The modified three-parameter linear Ross–Li BRDF model can be written as (Maignan et al. 2004):

$$r_{dd}(\theta_s,\theta_v,\phi) = f_{iso} + f_{vol}\cdot K_{vol}(\theta_s,\theta_v,\phi) + f_{geo}\cdot K_{geo}(\theta_s,\theta_v,\phi) \quad (1)$$

where the volumetric and geometrical kernel function has the following form:

$$K_{vol} = \frac{(\pi/2-\xi)\cos\xi + \sin\xi}{\cos\theta_s + \cos\theta_v}\left(1+\frac{1}{1+\xi/\xi_0}\right) - \frac{\pi}{4} \quad (2)$$

$$K_{geo} = O(\theta_s,\theta_v,\phi) - \sec\theta_s - \sec\theta_v + 0.5(1+\cos\xi)\sec\theta_s\sec\theta_v \quad (3)$$

and where

$$O = (t - \sin t \cos t)(\sec\theta_s + \sec\theta_v)/\pi \quad (4)$$

$$\cos t = \frac{h\sqrt{D^2 + (\tan\theta_s \tan\theta_v \sin\phi)^2}}{b(\sec\theta_s + \sec\theta_v)} \quad (5)$$

$$D = \sqrt{\tan^2\theta_s + \tan^2\theta_v - 2\tan\theta_s \tan\theta_v \cos\phi} \quad (6)$$

$$\cos\xi = \cos\theta_s \cos\theta_v + \sin\theta_s \sin\theta_v \cos\phi \quad (7)$$

and where ξ_0=0.026, $\frac{h}{b}$=2.0, and all the angles have the unit of radian.

3.4.1.1.2 Formulation of TOA reflectance

To retrieve AOD and the parameters of the surface BRDF from TOA reflectance, we have to establish TOA reflectance as a function of BRDF parameters and AOD. Here, we use the formulation proposed by Qin et al. (2001). The formula for TOA reflectance $\rho(\Omega_s, \Omega_v)$is expressed as:

$$\rho(\Omega_s, \Omega_v) = \rho_0(\Omega_s, \Omega_v) + \frac{T(\Omega_s)R(\Omega_s, \Omega_v)T(\Omega_v) - t_{dd}(\Omega_s)t_{dd}(\Omega_v)\left|R(\Omega_s, \Omega_v)\right|\bar{\rho}}{1 - r_{hh}\bar{\rho}} \quad (8)$$

where $\Omega_s \in (-\mu_s, \phi_s)$is the solar incoming direction, and $\Omega_v \in (\mu_v, \phi_v)$ for the viewing direction. There are two groups of coefficients in the above expression that are independent of each other: atmosphere-dependent and surface-dependent. These coefficients in each group represent the inherent properties of either the atmosphere or the surface. This means that we can determine these two groups of coefficients separately.

For the atmosphere, $\rho_0(\Omega_s, \Omega_v)$ is the atmospheric reflectance associated with path radiance (zero surface reflectance), and $\bar{\rho}$ is the atmospheric spherical albedo as defined before. The transmittance matrices are defined as:

$$T(\Omega_s) = \begin{bmatrix} t_{dd}(\Omega_s) & t_{dh}(\Omega_s) \end{bmatrix} \quad (9)$$

$$T(\Omega_v) = \begin{bmatrix} t_{dd}(\Omega_v) & t_{hd}(\Omega_v) \end{bmatrix}^T \quad (10)$$

where the subscript T stands for transpose, each transmittance has two subscript symbols: d (directional) and h (hemispherical).

The direct transmittance (t_{dd}) has the simple analytical expression: $t_{dd}(\mu) = \exp(-\tau_t/\mu)$.

The directional-hemispheric transmittance (t_{dh}) defines the fraction of downward diffuse flux generated by atmospheric scattering as the direct beam passes through the atmosphere. It can be calculated as the ratio of the integrated sky radiance at the surface level $L^{\downarrow}(\Omega_s, \Omega_v)$ over the downward hemisphere to the TOA incoming solar radiation:

$$t_{dh}(\Omega_s) = \frac{\int_{2\pi^-} L^{\downarrow}(\Omega_s, \Omega_v)\mu_v d\Omega_v}{\mu_s F_0} \tag{11}$$

The hemispheric-directional transmittance (t_{hd}) is defined as the ratio of the integrated upwelling TOA radiance over the upper hemisphere to the upwelling flux at the surface level $F^{\uparrow}$:

$$t_{hd}(\Omega_v) = \frac{\int_{2\pi^+} L^{\uparrow}(\Omega_s, \Omega_v)\mu_s d\Omega_s}{F^{\uparrow}}. \tag{12}$$

where both t_{dh} and t_{hd} have to be calculated numerically. A practical solution is to create look-up tables in advance.

For the surface, the reflectance matrix is defined as:

$$R(\Omega_s, \Omega_v) = \begin{bmatrix} r_{dd}(\Omega_s, \Omega_v) & r_{dh}(\Omega_s) \\ r_{hd}(\Omega_v) & r_{hh} \end{bmatrix} \tag{13}$$

where $r_{dd}(\Omega_s, \Omega_v)$ is the surface BRF. The directional-hemispherical reflectance $r_{dh}(\Omega_s)$ (or black-sky albedo) is defined as:

$$r_{dh}(\Omega_s) = \frac{1}{\pi}\int_{2\pi^+} r_{dd}(\Omega_s, \Omega_v) d\Omega_v, \tag{14}$$

where the integration is over the upper hemisphere. The hemispherical-directional reflectance $r_{hd}(\Omega_v)$ (or white-sky albedo) is defined in the same way, but the integration is over the lower hemisphere:

$$r_{hd}(\Omega_v) = \frac{1}{\pi}\int_{2\pi^-} r_{dd}(\Omega_s, \Omega_v) d\Omega_s \tag{15}$$

The bi-hemispherical reflectance (r_{hh}) is:

$$r_{hh} = 2\int_0^1 r_{dh}(\mu_s)\mu_s d\mu_s \tag{16}$$

where $\mu_s = \cos(\theta_s)$.

The determinant $|R|$ is easily calculated as:

$$|R(\Omega_s, \Omega_v)| = r_{dd}(\Omega_s, \Omega_v) r_{hh} - r_{dh}(\Omega_s) r_{hd}(\Omega_v) \tag{17}$$

It is evident that as long as surface BRDF parameters are known, the surface reflectance matrix can be determined. The authors claim that this approach does not introduce any approximation into the formulation, and their numerical experiments demonstrate that this formulation is very accurate (Qin et al., 2001).

3.4.1.1.3 Calculation of albedos

After obtaining BRDF parameters, it is straightforward to calculate spectral albedos, which are simply integrations of the surface directional reflectance functions over the entire viewing hemisphere. The spectral albedos are denoted as narrowband albedos in the next section. Instead of directly carrying out the numeric integration, we calculate the integral using an empirical polynomial equation of the three kernel parameters, similar to the MODIS albedo algorithm (Schaaf et al. 2002). The details are given in Section 3.5.4.
An angular integration over all the viewing angles is required to calculate albedo from BRDF parameters. Instead of directly calculating the integral, we use a similar method to MODIS (Schaaf et al. 2002), fitting blacksky albedo with a polynomial function:

$$\alpha_{bs}(\theta_s) = f_{iso}\,a + f_{vol}(b_0 + b_1\theta_s + b_2{\theta_s}^2 + b_3{\theta_s}^3) + f_{geo}(c_0 + c_1\theta_s + c_2{\theta_s}^2 + c_3{\theta_s}^3) \quad (18)$$

Where θ_s is the solar zenith angle, and $a_0, a_1, a_2, b_0, b_1, b_2, c_0, c_1, c_2$ are the regression coefficients, whose values are listed in Table 3.8. Similarly, the whitesky albedo can be computed by using the equation:

$$\alpha_{ws} = f_{iso}\,a + f_{vol}\,b + f_{geo}\,c \quad (19)$$

Table 3.8. Coefficients used to calculate albedo from BRDF parameters.

Variable	Value
a	1.0
b_0	-0.0374
b_1	0.5699
b_2	-1.1252
b_3	0.8432
c_0	-1.2665
c_1	-0.1662
c_2	0.1829
c_3	-0.1489
a	1.0
b	0.2260
c	-1.3763

After the narrowband albedos are obtained from the integration of the directional reflectance model, narrowband to broadband conversions are carried out based on empirical statistical relationships. The broadband albedo mainly depends on surface

spectral albedo spectra, but is also affected by the atmospheric conditions. With extensive radiative transfer simulations and surface reflectance spectral measurements, we have developed the conversion formulas for calculating the total shortwave albedo, total-, direct-, and diffuse-, visible, and near-infrared broadband albedos for several narrowband sensors (Liang 2001; Liang et al. 2003; Liang et al. 1999), including ASTER, AVHRR, GOES, Landsat-7 ETM+, MISR, MODIS, POLDER, and VEGETATION in SPOT spacecraft. A similar approach was later applied to generate the conversion formula for VIIRS (Liang et al. 2005a). The formula for MODIS has been used for routine albedo production (Schaaf et al. 2002), the MISR formula for calculating shortwave albedo is very effective (Chen et al. 2008), and the VIIRS formula will be used for operational albedo production. The same strategy also will be used to covert five ABI narrowband albedos to one broadband albedo.

The broadband albedo can be converted from spectral albedos using the following empirical formula:

$$\bar{r}(\theta_s) = \beta_0 + \beta_1 F_1 r_1(\theta_s) + \beta_2 F_2 r_2(\theta_s) + \beta_3 F_3 r_3(\theta_s) + \beta_5 F_5 r_5(\theta_s) + \beta_6 F_6 r_6(\theta_s) \quad (20)$$

where $r_i(\theta_s)$ are the spectral albedo, β_i are the coefficients, F_i are the normalized downward irradiance of the ABI five bands:

$$F_i = \frac{E_i(\theta_s)}{E_1(\theta_s) + E_2(\theta_s) + E_3(\theta_s) + E_5(\theta_s) + E_{6i}(\theta_s)} \quad (21)$$

and $E_i(\theta_s)$ are the downward irradiance of each band (at the specific solar zenith angle). Radiative transfer simulations and statistical analysis provide the coefficients β_i. The equations used for three sensors are given below:

$$\alpha_{MODIS} = 0.160\alpha_1 + 0.291\alpha_2 + 0.243\alpha_3 + 0.116\alpha_4 + 0.112\alpha_5 + 0.0713\alpha_7 - 0.0015$$

$$\alpha_{SEVIRI} = 0.4331\alpha_1 + 0.3939\alpha_2 + 0.1136\alpha_3 - 0.0084$$

$$\alpha_{ABI} = 0.2692\alpha_1 + 0.1661\alpha_2 + 0.3841\alpha_3 + 0.1138\alpha_5 + 0.0669\alpha_6$$

3.4.1.2 Derivation of BRDF parameters

Given the surface BRDF model (1) and the atmospheric radiative transfer model (8), the BRDF parameters and AOD at each observation can be obtained by minimizing the following cost function:

$$J(x) = (r(x) - r_b)B^{-1}(r(x) - r_b) + (\hat{\rho}(x) - \rho)R^{-1}(\hat{\rho}(x) - \rho) \quad (22)$$

where $\boldsymbol{x}$ are the three coefficients of the surface BRDF model and AOD, $r(x)$ is the calculated white-sky surface albedo using the BRDF model, r_b are the "first-guess" values of albedo from albedo climatology, B is the uncertainty matrix of the albedo "first-guess" values, ρ is the observed ABI TOA reflectance, $\hat{\rho}$ is the calculated TOA reflectance from equation (8), and R is the error matrix of the calculated TOA reflectance.

In this cost function, r_b is from the albedo climatology, B and R are preset values; ρ is the observed TOA reflectance. Given one group of BRDF parameters, the white-sky surface albedo could be calculated easily (see Section 3.4.1.1.3). Given AOD in addition to BRDF parameters, the TOA spectral reflectance could be calculated from the Qin's radiative transfer equation (8) (see Section 3.4.1.1.2). In this optimization system, the parameters to be optimized are spectral BRDF parameters and AOD. There are many different approaches available to minimize the cost function and obtain the BRDF parameters. We employ the Shuffled Complex Evolution method (SCE-UA, Duan et al. (1992) and Duan et al. (1993)), an efficient algorithm in searching global optimals. The SCE-UA method is capable of handling high parameter dimensionality and it does not rely on the availability of an explicit expression for the objective function or the derivatives.

3.4.2 The online mode

According to the results of two tests (whether the offline mode returns successfully and whether clear-sky observations are available during the 60-minutes' window), the online mode chooses various paths to calculate albedos and BRFs.

3.4.2.1 Calculation of albedos

3.4.2.1.1 Routine algorithm

If the offline mode returns successfully, the online mode uses the retrievals of BRDF parameters from the offline mode to calculate albedos. This is called the routine algorithm. Using equations (18) and (19), we can obtain both black-sky and white-sky spectral albedos. The broadband black-sky and white-sky albedos could then be derived from Equation (21). Given the black-sky and white-sky broadband albedo, the blue-sky broadband albedo α can be calculated by:

$$\alpha = p\alpha_{ws} + (1-p)\alpha_{bs} \qquad (23)$$

where p is the diffuse fraction of the total radiation. The fraction of direct irradiance is one parameters of the LUT. Given AOD, the corresponding p could be searched from the LUT. If no AOD value is available, the default value of 0.1 will be used in searching LUT.

3.4.2.1.2 Back-up algorithm

The routine optimization algorithm needs aggregation of multiple clear-sky observations within a relatively short time periods, during which the surface conditions keep relatively stable. Continuous cloud coverage and rapidly changing surface usually cause the failure of this algorithm. To handle these cases, we propose the direct estimation approach as the back-up algorithm (Liang 2003; Liang et al. 2002; Liang et al. 2005b). Unlike the optimization approach, the direct estimation approach uses only one set of clear-sky observations at one time as the input. Validation shows this approach could produce reliable albedos over both "dark" surfaces and "bright" surfaces.

The mathematical formulation of this approach is not complicated. The extensive simulations of atmospheric radiative transfer show there exists good relationship between the surface albedo and TOA spectral reflectance, if the observing geometry is fixed. Instead of deriving BRDF parameters first and then calculating albedo from the BRDF parameters, the direct estimation approach directly calculate the surface albedo from TOA signals.

For a given combination of solar zenith angle θ_s, view zenith angle θ_v and relative azimuth angle φ, surface albedo α could be calculated from TOA spectral reflectance $\rho_i(\theta_s,\theta_v,\varphi)$ at five bands (i=1,2,3,5,6) using a linear equation:

$$\alpha = a_0(\theta_s,\theta_v,\varphi) + \sum_{i=1,2,3,5,6} a_i(\theta_s,\theta_v,\varphi)\rho_i(\theta_s,\theta_v,\varphi) \qquad (24)$$

where $a_i(\theta_s,\theta_v,\varphi)$ (i=0,1,2,3,5,6) are preset coefficients dependent on viewing geometry.

The regression coefficients were derived offline through extensive simulations of atmospheric radiative transfer. Two groups of coefficients are stored to calculate white-sky albedo and black-sky albedo respectively.

3.4.2.1.3 Graceful degradation

If the offline mode returns successfully, instantaneous albedos could be calculated whether clear-sky observation is available or not. However, the back-up algorithm requires clear-sky observations as input. If no clear sky observation is available and no BRDF parameters are retrieved, we have to find alternative to predict albedo, instead of providing filling values. Yang et al. (Yang et al. 2008) gave an empirical equation to calculate black-sky albedo α_{bs} from white-sky albedo α_{ws} when solar zenith angle θ is known:

$$\alpha_{bs} = \alpha_{ws}\frac{1+1.14}{1+1.48\cos\theta} \qquad (25)$$

We use this equation to predict instantaneous albedo from the climatological white-sky albedo for the case no clear-sky observation is available and no BRDF parameter is retrieved.

3.4.2.2 Calculation of surface BRF

In the online mode, surface BRF products are produced every 60 minutes. ABI will obtain four sets of images within the 60-minutes window. Depending on the availability of BRDF parameters and the amount of clear-sky observation within the 60-minutes window, the calculation of surface BRF has several different paths:

1. When the routine offline mode succeeds,
a. and when clear-sky observation is available in the 60-minute window, BRF is retrieved from atmospheric correction with a BRDF model, using the latest clear-sky observation as input (algorithm R1);
b. and when no clear-sky observation is available in the 60-minute window, BRF is calculated from the BRDF parameters (algorithm R2);
2. When the routine offline mode fails,
a. and when clear-sky observation is available in the 60-minute window, BRF is retrieved by assuming the Lambertian surface with the latest clear-sky observation as input (algorithm R3);
b. and when no clear-sky observation is available in the 60-minute window, filling value is given.

3.4.2.2.1 R1: Atmospheric correction with BRDF model

When BRDF parameters are available, the directional-hemispheric, hemispheric-directional, hemispheric-hemispheric reflectance could be calculated from the BRDF model. Given the fact that AOD is known, the only unknown in equation (8) is the directional-directional reflectance (BRF). Thus, the surface reflectance could be calculated by inverting equation (8).

3.4.2.2.2 R2: Prediction from BRDF model

BRF could always be calculated from the BRDF model (1), given BRDF parameters and viewing geometry. Numerically, BRF has the following relationship with BRDF:

$$BRF = BRDF * \pi \quad (26)$$

However, the retrieved BRDF parameters from the offline mode represent average surface conditions of the compositing time period. Large uncertainties exist when we predict instantaneous surface BRF from the offline BRDF parameters. This simple calculation is used when no clear-sky observation is available.

3.4.2.2.3 R3: Lambertian correction

When no BRDF parameters are available, we assume the surface is Lambertian and obtain a simplified form of equation (8):

$$r = r_0 + \frac{r_s}{1 - r_s \rho} \gamma \quad (27)$$

where r is TOA reflectance, r_s is surface reflectance, ρ is spherical albedo, r_0 is path reflectance, and γ is transmittance. Given viewing geometry and AOD, ρ, r_0 and γ could be obtained from the LUT. Thus, the surface reflectance could be solved from the following equation:

$$r_s = \frac{r - r_0}{\gamma + (r - r_0)\rho} \quad (28)$$

3.5 Algorithm Output

The outputs of the LSA algorithm offline mode mainly include the three parameters of the BRDF model for each ABI band (Table 3.9), which will be used as one input of the online mode. The final outputs of the LSA algorithm online mode are instantaneous LSA and BRFs (Table 3.10 and 3.11).

Table 3.9. Outputs of the ABI albedo algorithm offline mode.

Name	Type	Description	Dimension
Ch1 f_iso	float	BRDF isotropic component parameter at Ch1	grid (xsize, ysize)
Ch1 f_vol	float	BRDF volumetric kernel parameter at Ch1	grid (xsize, ysize)
Ch1 f_geo	float	BRDF geometric kernel parameter at Ch1	grid (xsize, ysize)
Ch2 f_iso	float	BRDF isotropic component parameter at Ch2	grid (xsize, ysize)
Ch2	float	BRDF volumetric kernel parameter at	grid (xsize,

f_vol		Ch2	ysize)
Ch2 f_geo	float	BRDF geometric kernel parameter at Ch2	grid (xsize, ysize)
Ch3 f_iso	float	BRDF isotropic component parameter at Ch3	grid (xsize, ysize)
Ch3 f_vol	float	BRDF volumetric kernel parameter at Ch3	grid (xsize, ysize)
Ch3 f_geo	float	BRDF geometric kernel parameter at Ch3	grid (xsize, ysize)
Ch5 f_iso	float	BRDF isotropic component parameter at Ch5	grid (xsize, ysize)
Ch5 f_vol	float	BRDF volumetric kernel parameter at Ch5	grid (xsize, ysize)
Ch5 f_geo	float	BRDF geometric kernel parameter at Ch5	grid (xsize, ysize)
Ch6 f_iso	float	BRDF isotropic component parameter at Ch6	grid (xsize, ysize)
Ch6 f_vol	float	BRDF volumetric kernel parameter at Ch6	grid (xsize, ysize)
Ch6 f_geo	float	BRDF geometric kernel parameter at Ch6	grid (xsize, ysize)
QF	char	Quality flag for each pixel, indicating the general retrieval quality	grid (xsize, ysize)

Table 3.10. Outputs of the ABI albedo algorithm online mode: LSA

Name	Type	Description	Dimension
Shortwave Albedo	float	Derived broadband albedo value at 0.4-3.0 μm	grid (xsize, ysize)
Albedo QF	char	Quality flag for each pixel, indicating the general retrieval quality	grid (xsize, ysize)

Table 3.11. Outputs of the ABI albedo algorithm online mode: BRF

Name	Type	Description	Dimension
Ch1 BRF	float	Derived bidirectional reflectance value at 0.47 μm	grid (xsize, ysize)
Ch2 BRF	float	Derived bidirectional reflectance value at 0.64 μm	grid (xsize, ysize)
Ch3 BRF	float	Derived bidirectional reflectance value at 0.86 μm	grid (xsize, ysize)
Ch5 BRF	float	Derived bidirectional reflectance value at 1.61 μm	grid (xsize, ysize)

Ch6 BRF	float	Derived bidirectional reflectance value at 2.26 μm	grid (xsize, ysize)

The GOES-R ABI LSA and BRF products are generated from multiple paths with various levels of uncertainties. Detailed quality information is critical to the end users. Correspondingly, we also have three groups of Quality Flag (QF). One is for the intermediate BRDF parameter products, the second group is for albedo products, and the third group is for reflectance products. The three groups of QF are defined in Tables 3.12-14.

Table 3.12. QF definition of ABI intermediate BRDF parameter products

Bit	Name	Value
0	Land mask	0:land, 1: water
1	Retrieval successful	0:BRDF successfully retrieved 1:Routine algorithm fails
2	Empty	Reserved for future usage
3		
4		
5		
6		
7		

Table 3.13. QF definition of ABI LSA products

Bit	Name	Value
0	Land mask	0:land, 1: water
1	SZA	0:SZA<67, 1: SZA>=67
2	LZA	0:LZA<70, 1: LZA>=70
3	Retrieval path	00:Routine algorithm, 01:Back-up algorithm, 10: Graceful degradation,11: No retrieval
4		
5	Additional info	Used for routine algorithm only 0: valid AOD or COD, 1: AOD/COD climatology
6		
7	Empty	Reserved for future usage

Table 3.14. QF definition of ABI BRF products

Bit	Name	Value
0	Land mask	0:land, 1: water

1	SZA	0:SZA<67, 1: SZA>=67
2	LZA	0:LZA<70, 1: LZA>=70
3	Retrieval path	00:R1, 01:R2, 10: R3,11: No retrieval
4		
5	Empty	Reserved for future usage
6		
7		

Besides the QF info, each file of BRDF parameters, LSA and BRF products also comes with metadata information. The metadata information is given in Tables 3.15-17.

Table 3.15. Metadata of ABI intermediate BRDF parameter products

Metadata	Source	Definition
Date	common	Beginning and end dates of the product
Time	common	Beginning and end times of the product
Dimension	common	Number of rows, number of columns
Product Name	common	The ABI land surface BRDF parameter product
Satellite	common	GOES-R satellite name
Instrument	common	ABI
Version	common	Product version number
Data type	BRDF	Data type used to store BRDF parameters
Scale	BRDF	Scale used to stretch BRDF parameters
Offset	BRDF	Offset used to stretch BRDF parameters
Filling Value	BRDF	Value representing no data produced
Product Unit	BRDF	BRDF parameters are dimensionless
Compositing Period	BRDF	The compositing time period used to derive BRDF parameters.
QF categories	BRDF	Number of QF flag values
QF percentages	BRDF	Percent of retrievals with each QF flag value

Table 3.16. Metadata of ABI LSA products

Metadata	Source	Definition
Date	common	Beginning and end dates of the product
Time	common	Beginning and end times of the product
Dimension	common	Number of rows, number of columns
Product	common	The ABI land surface albedo product

Name		
Satellite	common	GOES-R satellite name
Instrument	common	ABI
Version	common	Product version number
Data type	LSA	Data type used to store albedo values
Scale	LSA	Scale used to stretch albedo
Offset	LSA	Offset used to stretch albedo
Filling Value	LSA	Value representing no data produced
Valid Range	LSA	Valid range of albedo values, 0-1
Product Unit	LSA	Albedo is dimensionless
Compositing Period	LSA	The compositing time period used to derive BRDF parameters.
Statistics	LSA	Maximums, minimums, means and standard deviations of albedos
QF categories	LSA	Number of QF flag values
QF percentages	LSA	Percent of retrievals with each QF flag value

Table 3.17. Metadata of ABI BRF products

Metadata	Source	Definition
Date	common	Beginning and end dates of the product
Time	common	Beginning and end times of the product
Dimension	common	Number of rows, number of columns
Product Name	common	The ABI land surface reflectance product
Satellite	common	GOES-R satellite name
Instrument	common	ABI
Version	common	Product version number
Data type	BRF	Data type used to store reflectance values
Scale	BRF	Scale used to stretch reflectance
Offset	BRF	Offset used to stretch reflectance
Filling Value	BRF	Value representing no data produced
Valid Range	BRF	Valid range of reflectance values, 0-2
Product Unit	BRF	Reflectance is dimensionless
Compositing Period	BRF	The compositing time period used to derive BRDF parameters.
Statistics	BRF	Maximums, minimums, means and standard deviations of reflectances
QF categories	BRF	Number of QA flag values
QF	BRF	Percent of retrievals with each QF flag value

percentages		

4 TEST DATA SETS AND OUTPUTS

The algorithm will be tested using two types of proxy data: MODIS and the simulated ABI data. Both LSA products and BRF products have been validated.

4.1 Input Data Sets and Ground Measurements

4.1.1 Proxy Input Data

One characteristic of the ABI LSA algorithm is that it takes advantage of two features of ABI measurements -- high temporal refreshing rates and multi-spectral configuration -- to achieve the goal of retrieving atmospheric conditions and surface BRDF parameters simultaneously. MODIS has similar spectral configuration (seven bands for the land application) to ABI. However, MODIS is a polar-orbiting sensor and maps the Earth surface only twice in most cases even if we combine both Terra and Aqua satellites. Table 4.1 lists the visible and near infrared bands of the MODIS and ABI. Since no existing satellite sensors can provide ideal proxy data with such characteristics to facilitate our algorithm verification activities, we also carry out the simulation of radiative transfer and use simulated ABI data to test the LSA algorithm.

Table 4.1. Comparison of MODIS and ABI reflective bands.

Sensor	Temporal resolution	Channel No.	Wavelength Center (μm)	Bandwidth (μm)
ABI	15min	1	0.47	0.45 – 0.49
		2	0.64	0.59 – 0.69
		3	0.86	0.85 – 0.88
		5	1.61	1.58 – 1.64
		6	2.26	2.23 – 2.28
MODIS	Polar-orbitting	1	0.646	0.62-0.67
		2	0.857	0.84-0.88
		3	0.466	0.46-0.48
		4	0.554	0.55-0.57
		5	1.242	1.23-1.25
		6	1.629	1.63-1.65
		7	2.114	2.11-2.16

4.1.1.1 Simulated Data

Since none of existing satellite sensors can provide proxy data with both the high temporal resolution and multiple-channels as ABI data, simulation will be the only way to test our

algorithm's ability to handle ABI data before GOES-R is launched. We use Qin et al. (2001)'s formulation of atmospheric radiative transfer to simulate TOA signals and assure the simulated TOA signals have the following properties:

- Use the ABI band configuration and band response functions.
- Bear the realistic observing geometry and refreshing rate of ABI.
- Consider the couple between the atmosphere and surface BRFs.

In order to simulate the signals received by the spaceborne sensors, both the surface properties and atmospheric parameters are needed in addition to the sensor response functions. We use the BRDF parameters from the MODIS albedo products as the input of surface properties. The AOD data come from the field measurements at SURFRAD sites as the atmospheric conditions. Similar to the solar radiation, AOD is also measured every 3 minutes using visible Multi-Filter Rotating Shadowband Radiometers (MFRSR). SURFRAD AOD measurements include five bands (415.0, 501.6, 613.5, 671.7 and 867.5nm). However, the 6S simulation requires the AOD input at 550nm. We calculate this value using the Angstrom Equation:

$$\tau_\lambda = \beta\lambda^{-\alpha}$$

where λ is the wavelength, τ_λ is the AOD at λ. α and β are coefficients, which are obtained through the linear regression if AOD measurements at three bands are valid. An example of such time series of AOD at 550nm is given in Figure 4.1.

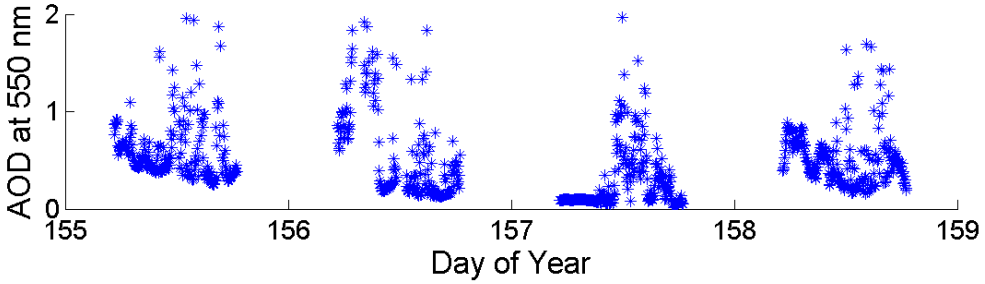


Figure 4.1 An example input of AOD time series at the Bondville site. The AODs at 550nm are calculated from AODs at other bands using the Angstrom Equation.

4.1.1.2 MODIS Data

We carried out two types of validation activities using MODIS as proxy data. First, we extracted the point data over a couple of sites, SURFRAD, AmeriFlux and GC-Net sites for the whole year of 2005 and directly compare our retrievals with field measurements. We also generated a time series of ten days' MODIS data as proxy data to test the operational ability of the LSA algorithm over 2D images. MODIS is a polar-orbiting sensor. The L1B MODIS swath cannot be directly used to stack the time series because each of these swaths has different spatial coverage. We have to project all the MODIS L1B swath data to a common map projection and stack them as time series data.

4.1.2 Ground Measurements

4.1.2.1 Measurement of albedo

Albedo measurements over SURFRAD, AmeriFlux and GC-Net networks are used in our validation (Table 4.2-4). Albedo is calculated as the ratio of outgoing and incoming solar irradiance. Incoming and outgoing shortwave radiation is measured every 3 minutes at SURFRAD sites using the Eppley Precision Spectral Pyranometers (PSP), which are calibrated annually. The averages of albedo over 30 minutes are used to compare with instantaneous albedo retrieved by our algorithm to mitigate the mismatch.

As many of the AmeriFlux sites do not have measurements on both shortwave upward and downward radiation, the visible radiation measurements are used in this study. Unlike the SURFRAD sites, the AmeriFlux sites refresh their measurements only every half an hour, so that a one-hour temporal window is used to calculate the "ground truth" blue-sky albedo similar to the SURFRAD data.

Ground radiation measurements over Greenland are regularly collected by the Greenland Climate Network (GC-Net). This data set provides unique and extensive observations which can help verify the validity of this proposed algorithm over snow-covered surfaces. Shortwave upward and downward radiation is observed on an hourly basis. The "ground truth" blue-sky albedo is calculated based on that. Thirteen sites were chosen in this study according to data availability and data quality during the year of 2003.

Table 4.2. Information of SURFRAD Stations

Site No.	Site Location	Latitude	Longitude	Surface types
1	Bondville, IL	40.05	-88.37	Crop
2	Desert Rock, NV	36.63	-116.02	Open shrub
3	Fort Peck, MT	48.31	-105.10	Grass
4	Goodwin Creek, MS	34.25	-89.87	Deciduous Forest
5	Pennsylvania State University, PA	40.72	-77.93	Mixed Forest
6	Sioux Falls, SD	43.73	-96.62	Forest

Table 4.3. Information of AmeriFlux Stations

Site No.	Site Location	Latitude	Longitude

1	Fort Peck	48.31	-105.10
2	Fermi(Prairie)	41.84	-88.24
3	Mead(Irrigated)	41.17	-96.48
4	Mead(Rain fed)	41.18	-96.44

Table 4.4. Information of GC-Net Stations

Site Name	Location	Site Name	Location
Swiss Camp	69.5732N, 49.2952W	CP1	69.8819N, 46.9736W
JAR1	69.4981N, 49.6816W	NASA-U	73.8333N, 49.4953W
JAR3	69.3954N, 50.3104W	GITS	77.1433N, 61.0950W
Summit	72.5794N, 38.5042W	Tunu-N	78.0168N, 33.9939W
Saddle	66.0006N, 44.5014W	DYE-2	66.4810, 46.2800W
NASA-SE	66.4797N, 42.5002W	NASA-E	75.0000N, 29.9997W
NGRIP	75.0998N, 42.3326W		

4.1.2.2 Surrogate of surface reflectance

Due to the limited spatial representation of ground measurements, it is always difficult to validate satellite pixel-based surface albedo estimations solely by comparing with ground measured data, especially when the pixel is not quite homogeneous. Using other satellite-derived data sources can help verify the algorithm estimations. Based on the ancillary information on aerosol and water vapor from the Aerosol Robotic Network (AERONET) sites, a set of surface albedo and reflectance data is retrieved through an independent atmospheric correction with the Ross-Li BRDF kernel models using TOA data from MODIS observations (Wang et al. 2009). MODASRVN data products from 2000 onwards are stored with the AERONET site in the center of the image covering 50 × 50 km2 at 1 km resolution.

According to the location, land cover type, and MODASRVN data availability from the AERONET sites, sixteen sites were chosen in this study for the validation of the estimated surface reflectance (Table 4.5). Similar to the ground measurement section, data for the year of 2005 for MODASRVN and MODIS level 1B TOA observations were collected and processed.

Table 4.5. MODASRVN – AERONET site information

Site Name	Location	Land Cover	Site Name	Location	Land Cover
Bondville	40.053 N, 88.372 W	Crop	Mexico City	19.334 N, 99.182 W	Urban
GSFC	38.993	Forest	Rimrock	46.487	Gras

	N, 76.840 W	& Urban		N, 116.990 W	s
Missoula	46.917 N, 114.080 W	Grass & Urban	MD Science Center	39.283 N, 76.617 W	Urban
SERC	38.883 N, 76.500 W	Forest & Wetland	KONZAEDC	39.102 N, 96.610 W	Grass
CARTEL	45.379 N, 71.931 W	Grass & Urban	BSRNBAO Boulder	40.045 N, 105.010 W	Grass
Bratts Lake	50.280 N, 104.700 W	Crop	Railroad Valley	38.504 N, 115.960 W	Grass
Sioux Falls	43.736 N, 96.626 W	Grass	Frenso	36.782 N, 119.770 W	Urban
Egbert	44.226 N, 79.750 W	Crop	Halifax	44.638 N, 63.594 W	Urban

4.2 Validation Results

4.2.1 Output from Simulated Data

More validation work using simulated ABI data is ongoing. Preliminary results show our retrieval could capture the BRDF distribution well under a variety of atmospheric and surface settings. One example is given in Figure 4.2. Although the two BRDF data sets come from different empirical models, they have similar reflectance distribution shapes. However, due to the feature of the BRDF model we use in the albedo algorithm, a hot-spot effect is noticeable in our retrieved BRDF distribution.

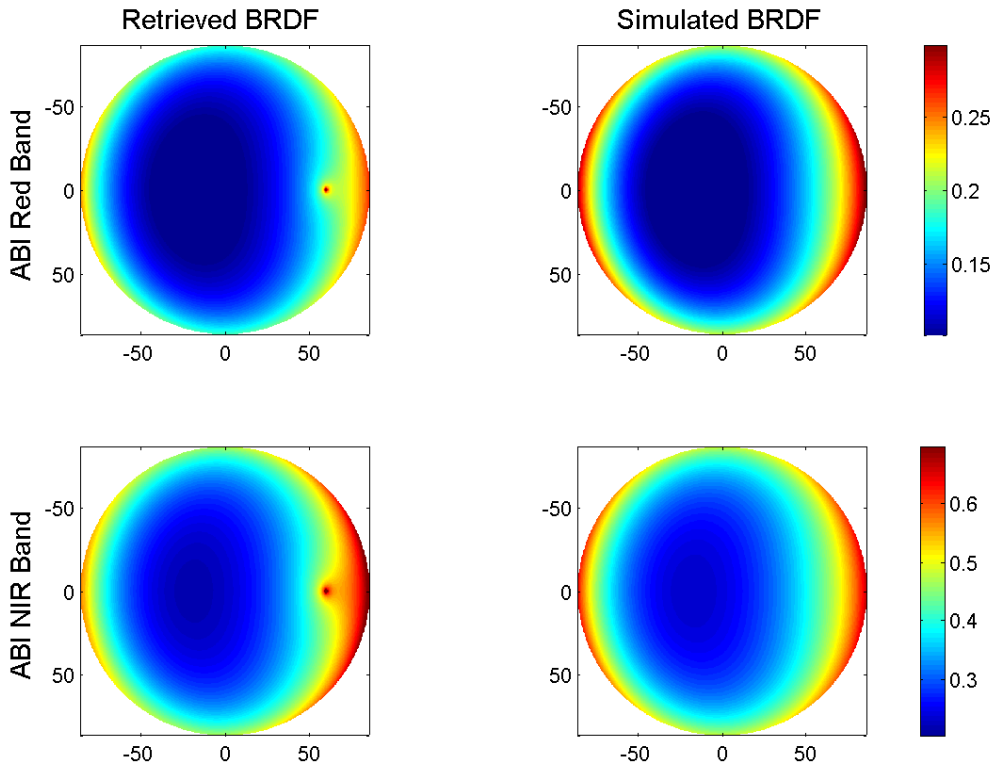
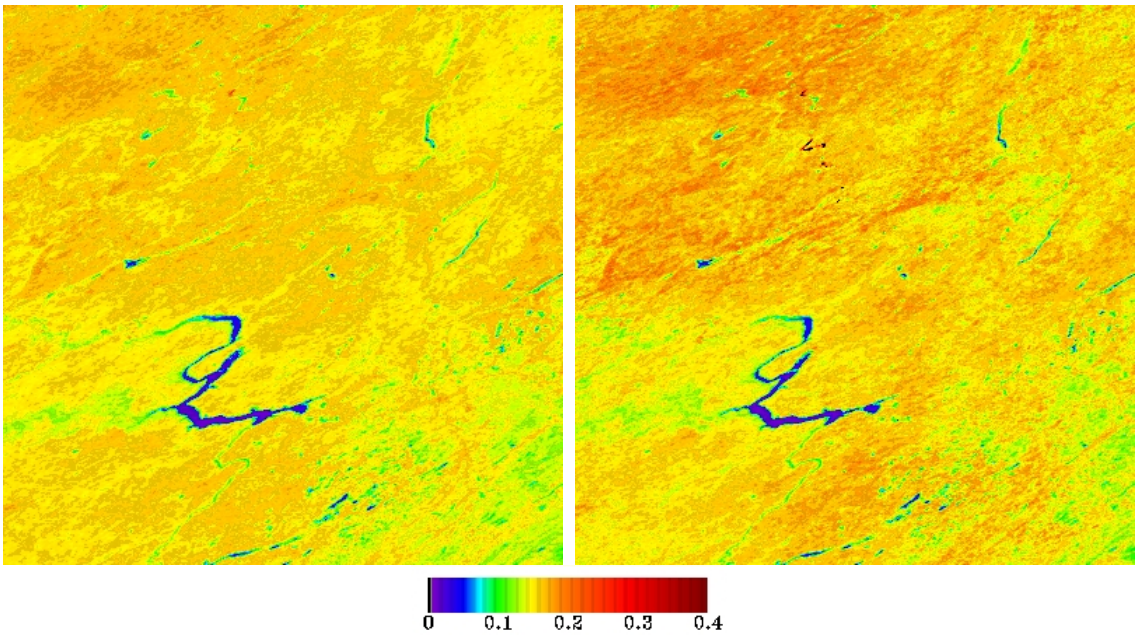


Figure 4.2 The retrieved BRDF and the actual BRDF used in the simulation at ABI red and near infrared bands.

4.2.2 Output from MODIS Data

The LSA algorithm was carried out on these time series of images to generate albedo maps at each observation time. The retrieved blacksky albedo on May 1st, 2005 around 48.3˚N, 102.8˚W is shown in Figure 4.3. The MODIS blacksky albedo for the same time and location is also shown. Our estimation captures a similar spatial pattern to MODIS data but has slight underestimation (Figure 4.4).



a) Estimated Blacksky Albedo b) MODIS Blacksky Albedo

Figure 4.3 The blacksky albedo maps on May 1st, 2005 around 48.3˚N, 102.8˚W.

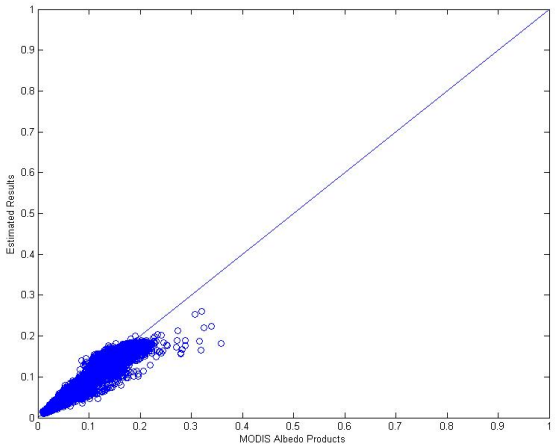
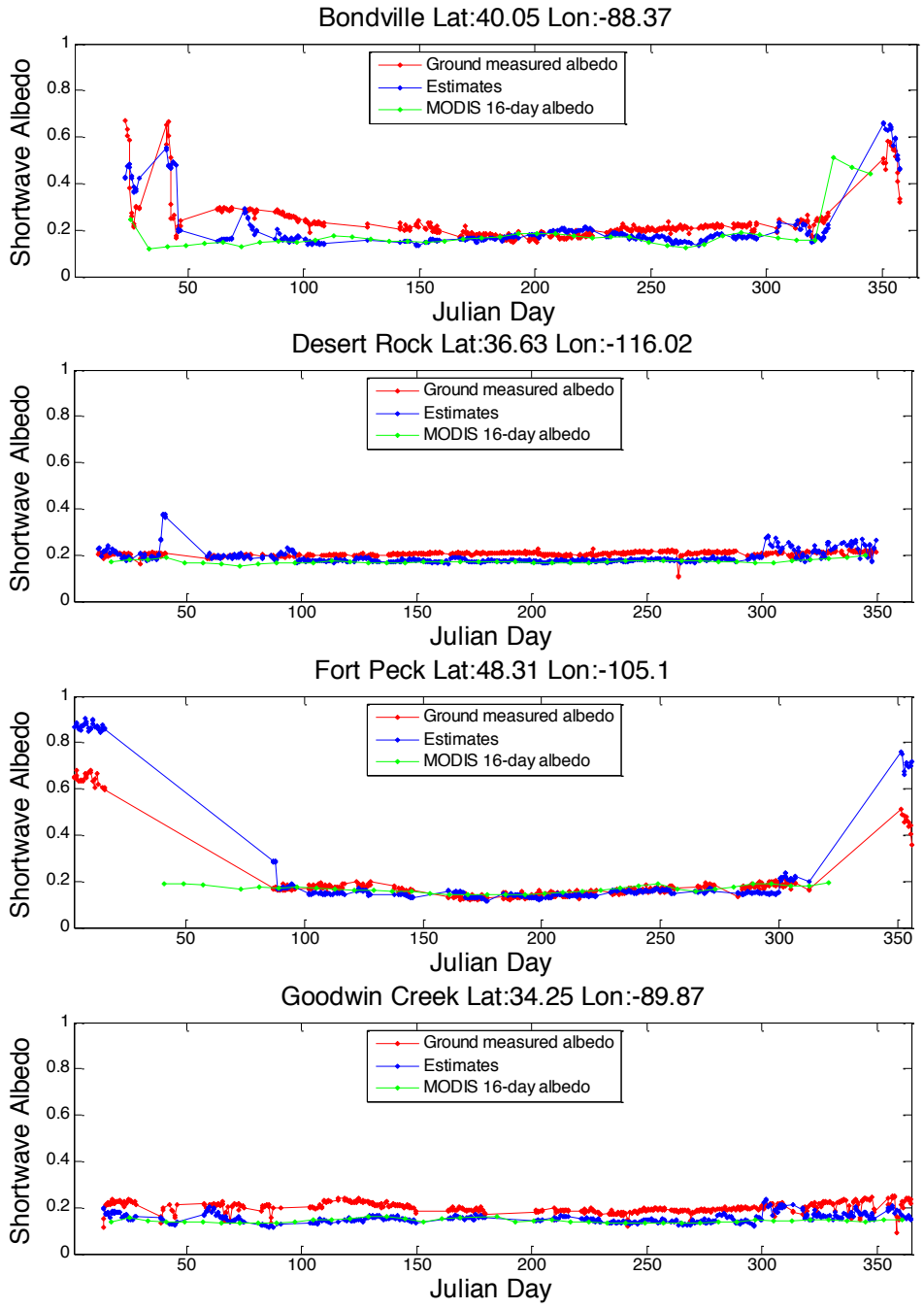


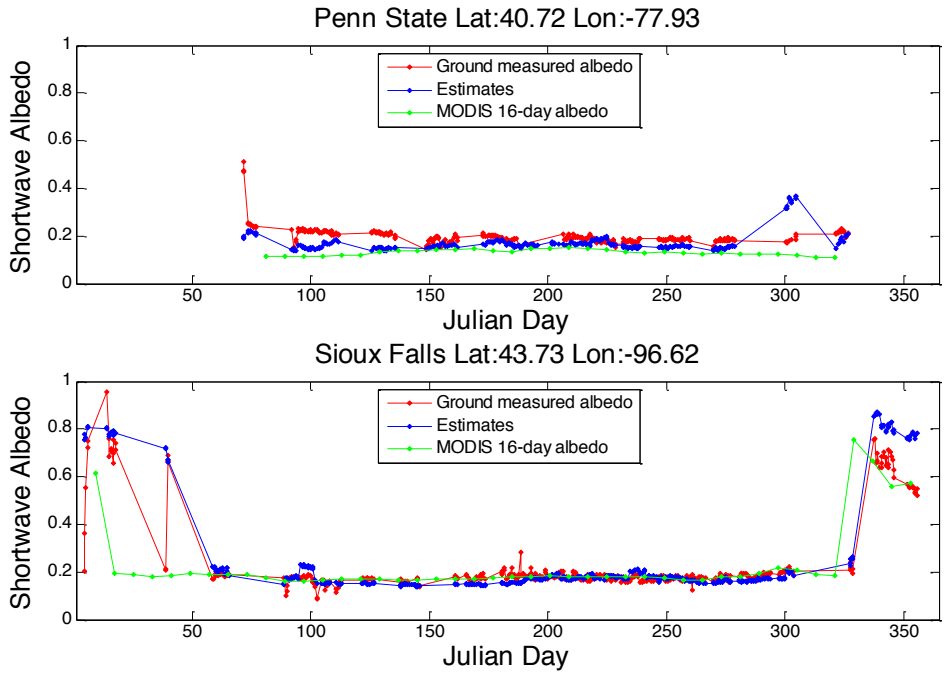
Figure 4.4 Comparison between our retrieved albedo and MODIS albedo.

4.2.3 Validation results of albedo

The direct validation results of our retrieved albedo values over three networks are shown in time series (Figure 4.5). Generally, the retrieved albedo values match well with field measurements. For the non-snow cases (Desert Rock and Goodwin Creek), the Root Mean Square Errors (RMSE) are quite small, although the R^2 values are rather low due to the small range of albedo variations. The undetected clouds may be the main cause of albedo overestimation in Desert

Rock. At Goodwin Creek, both our estimations and MODIS products are slightly lower than field measurements. This may come from the inaccurate representations of aerosol types. Both our retrievals and MODIS albedo data can reasonably represent the seasonal snow albedo over Bondville, Fort Peck and Sioux Falls. Our retrieval algorithm requires shorter compositing window, which makes it possible to better capture the rapid snowfall/melting processes when ABI data are available.



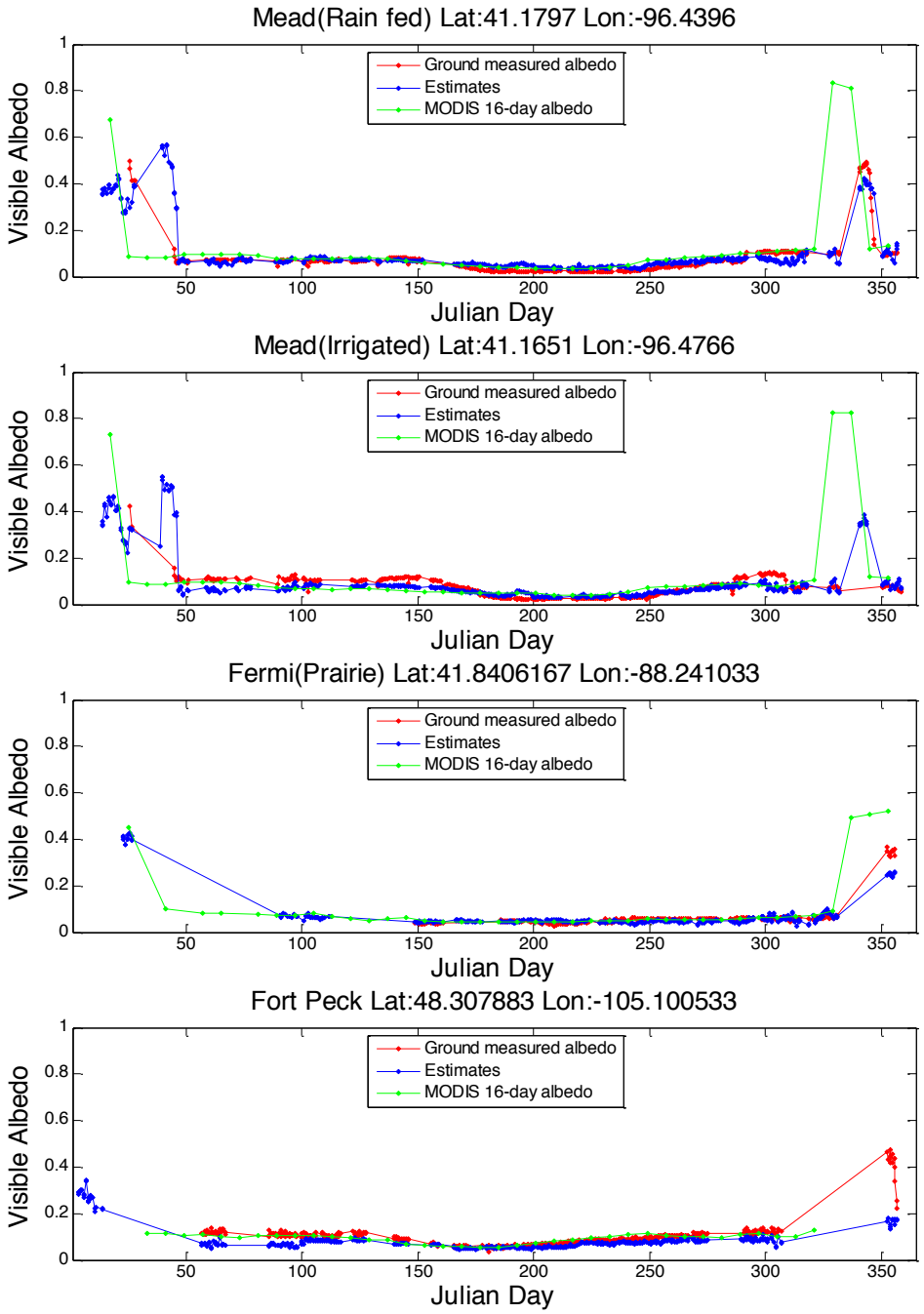


Error! Reference source not found.

Table 4.6. Statistics of the retrieved values from this study and MODIS albedo products with comparison to ground measurements over SURFRAD sites

	Bias	STD	RMSE	R^2
Bondville	0.003	0.065	0.071	0.625
Desert Rock	0.014	0.029	0.033	0.006
Fort Peck	-0.016	0.075	0.077	0.969
Goodwin Creek	0.046	0.026	0.053	0.051
Penn State	0.029	0.049	0.056	0.007
Sioux Falls	-0.016	0.072	0.074	0.907
All sites	0.014	0.060	0.062	0.821
All sites from MODIS	0.041	0.055	0.068	0.654

The validation results at four AmeriFlux sites are shown in Figures 4.6. The results shown here are visible albedo because only irradiances in the visible spectrum are measured at these sites. Similar to the results at SURFRAD, our algorithm can capture the annual curves of albedo, but produces larger errors for rapidly changing surfaces. Furthermore, the MODIS albedo products have several gaps when snow is found over some sites or simply provide snow-free albedo estimations if either there are not enough observations for retrieving or only data with low accuracy can be obtained.



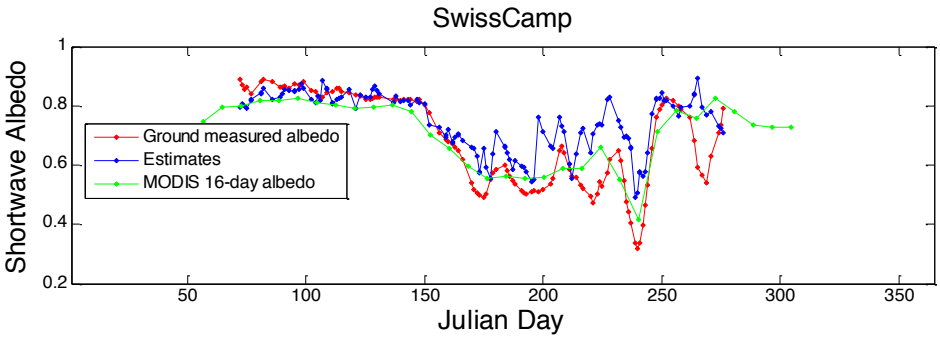
Error! Reference source not found.

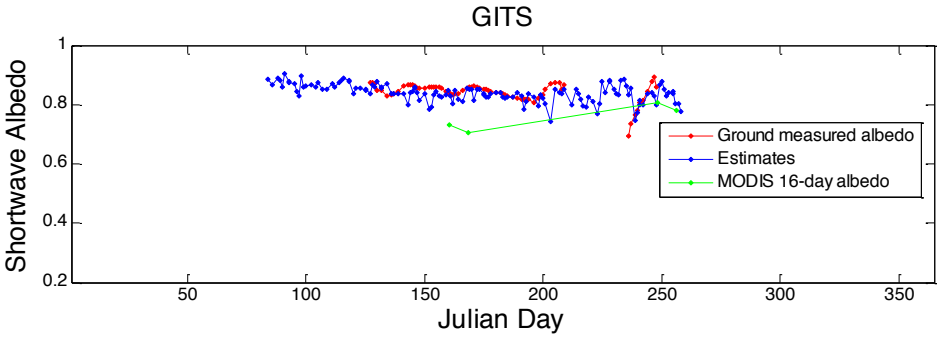
Table 4.7. Statistics of the retrieved values from this study and MODIS albedo products with comparison to ground measurements over AmeriFlux sites

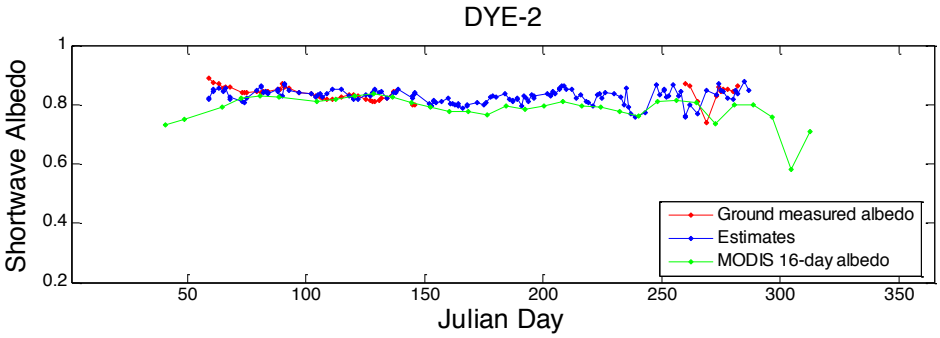
	Bias	STD	RMSE	R^2
Mead (Rain fed)	-0.007	0.038	0.039	0.769
Mead (Irrigated)	0.005	0.036	0.037	0.371
Fermi	0.002	0.021	0.021	0.886

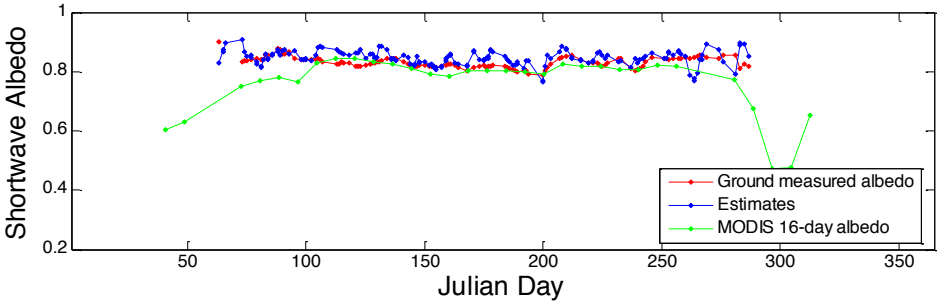
Fort Peck	0.024	0.044	0.050	0.479
All sites	0.006	0.386	0.039	0.591
All sites from MODIS	-0.007	0.107	0.107	0.701

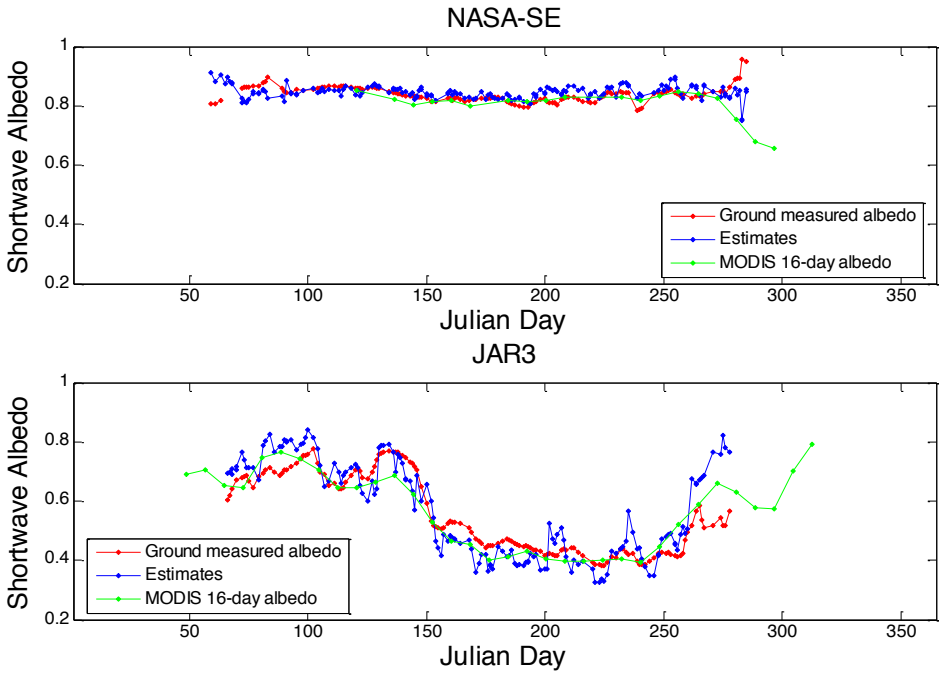
Time series comparisons of ground measurements, retrieved albedo values, and MODIS albedo products over the GC-Net sites are given in Figure 4.7. From the results shown here, snow and snow-melt events were clearly captured by the retrievals of this proposed algorithm. Variations of ground measurements and the retrieved albedo data can be found on the daily basis results whereas the 16-day MODIS albedo curves are smooth over most cases. The cause of the albedo variations is the changes in solar zenith angle, since MODIS can have multiple overpasses over Greenland in one day (combined Terra and Aqua). As more observations can be obtained over Greenland compared to those of the SURFRAD and AmeriFlux sites, the time range of collecting the cloud free observations is actually shorter over the Greenland sites, which gives the algorithm better capability capturing rapid changes. This algorithm gave a satisfying result over all sites with a small positive bias (0.012). The overall R2 (0.842) shows that the albedo retrievals have a good correlation with the ground measurements indicating that the changes of snow surfaces can be well captured, although sometimes the sliding window size is still larger than the real situation given that the RMSEs are higher than 0.05 over some sites.


SwissCamp
Shortwave Albedo
Julian Day
Ground measured albedo
Estimates
MODIS 16-day albedo


GITS
Shortwave Albedo
Julian Day
Ground measured albedo
Estimates
MODIS 16-day albedo


DYE-2
Shortwave Albedo
Julian Day
Ground measured albedo
Estimates
MODIS 16-day albedo


Saddle
Shortwave Albedo
Julian Day
Ground measured albedo
Estimates
MODIS 16-day albedo



Error! Reference source not found.

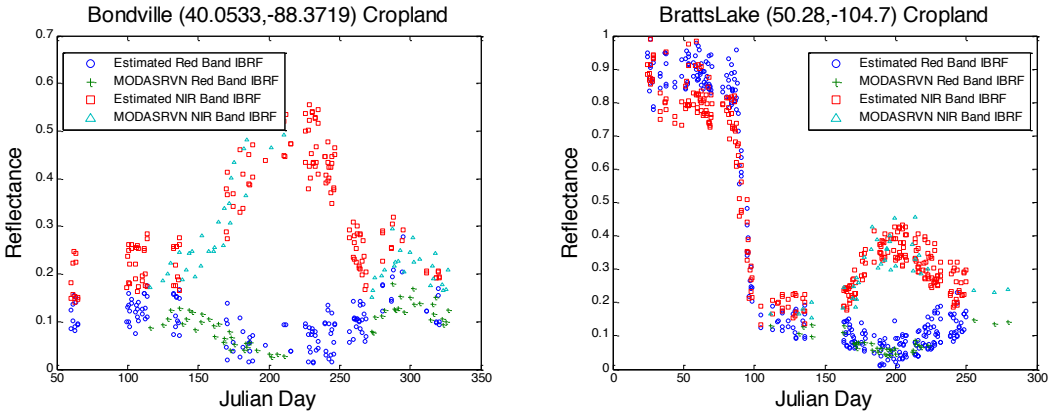
Table 4.8. Statistics of the retrieved values from this study comparison to ground measurements over GC-Net sites

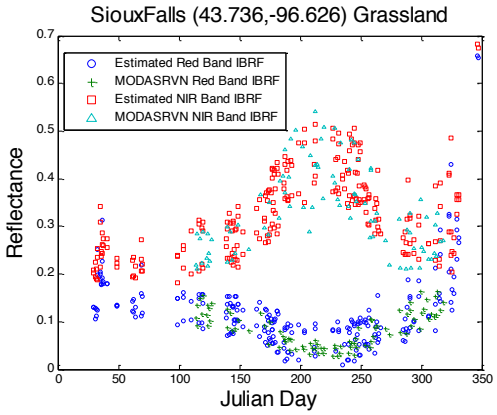
	Bias	RMSE	R^2
Swiss Camp	0.062	0.111	0.700
GITS	-0.011	0.036	0.077
Summit	0.010	0.047	0.027
DYE-2	-0.004	0.033	0.006
JAR1	0.015	0.091	0.872
Saddle	0.015	0.030	0.060
NASA-E	-0.024	0.034	0.002
NASA-SE	0.007	0.035	0.000
JAR3	0.001	0.077	0.774
All sites	0.012	0.065	0.842

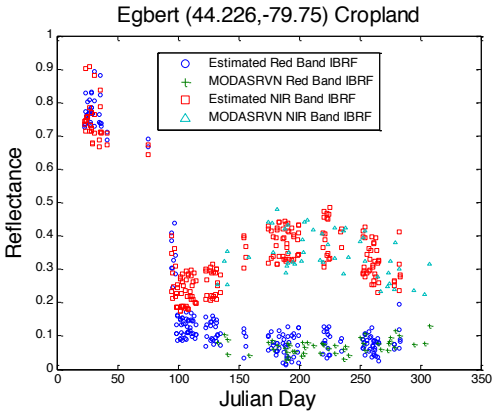
4.2.4 Validation results of BRF

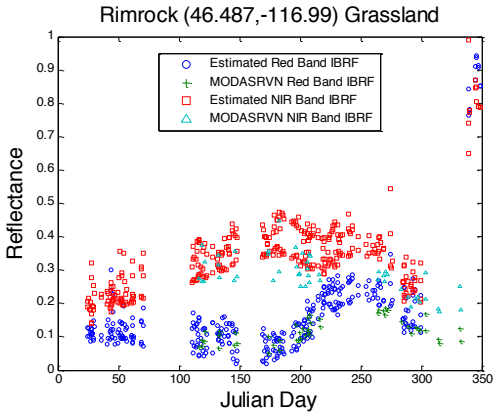
Sixteen sites are chosen in the validation of the surface reflectance using the MODASRVN data set. Time-series comparisons of the red band and near-infrared band data over six vegetation sites are given in Figure 4.8. The retrieved surface reflectances in

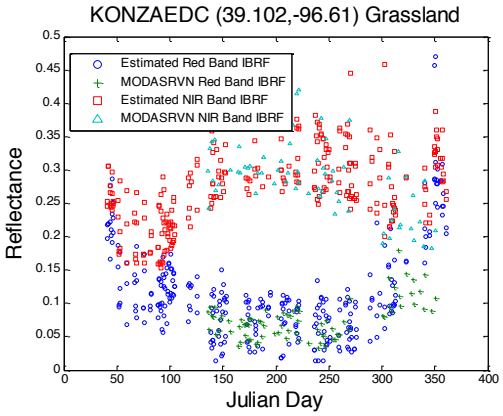
the two bands capture the seasonal trends and match the MODASRVN instantaneous reflectance products very well. However, due to the failure to provide the reflectance over the snow covered surfaces from the MODASRVN data in most cases, it is difficult to validate the proposed algorithm over bright surfaces using this dataset. Moreover, as this dataset only relies on the MODIS sensor onboard Terra, fewer retrievals are available than our results in this study. Direct comparisons are given in Figure 4.9 over all sixteen sites for all 7 MODIS bands. The overall correlation of the retrievals and MODASRVN data is very good for each individual band and the bias and RMSE are small. The R-squared values are relatively small for band 3 and band 4 as there is only a narrow range for the reflectances (0–0.2). Some outliers are found in the comparison, probably due to the misclassification of the cloud mask, which is one of the major input components for this algorithm. Given the variability of surface cover types in all the sixteen sites, the results show that the algorithm proposed here is capable of handling different types of land cover regardless of its homogeneity.



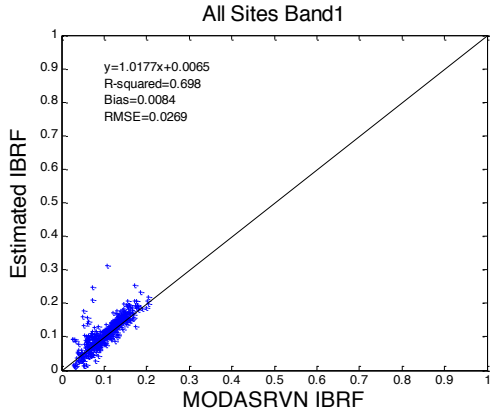


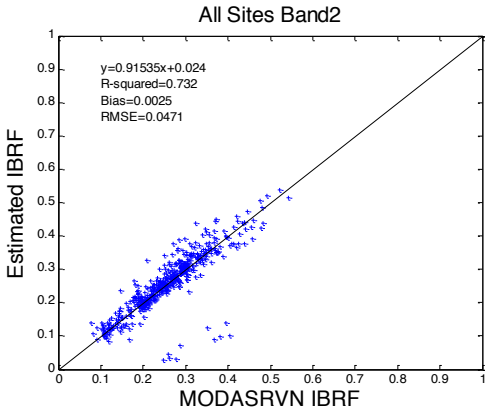




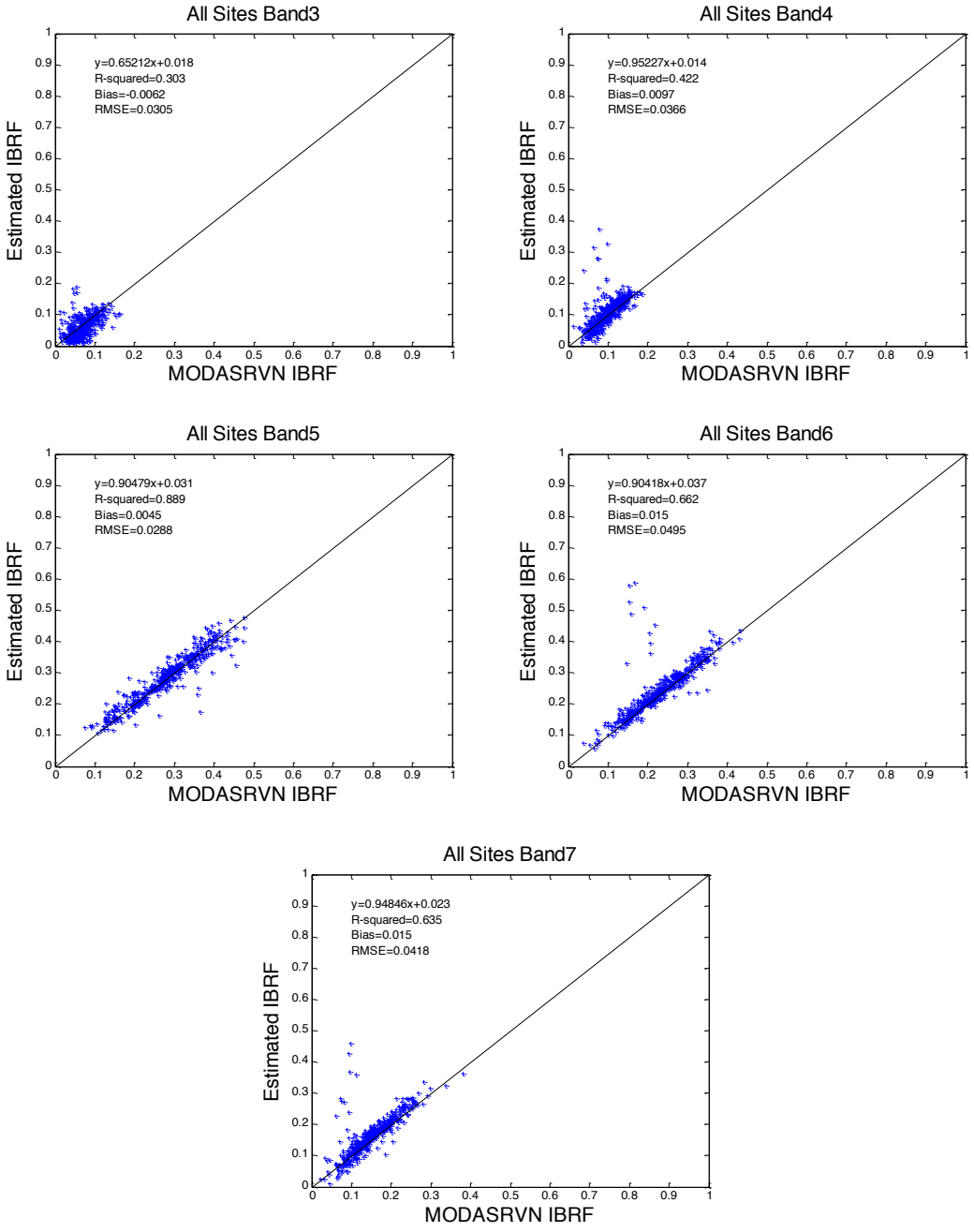


Error! Reference source not found.





.



Error! Reference source not found.

Table 4.9. Statistics of the retrieved reflectance values from this study with comparison to MODASRVN reflectance products ground measurements over sixteen AERONET sites

Band No.	Bias	RMSE	R^2
1	0.008	0.027	0.698
2	0.003	0.047	0.732
3	-0.006	0.031	0.303
4	0.010	0.037	0.422

5	0.005	0.029	0.889
6	0.015	0.050	0.662
7	0.015	0.042	0.635

4.2.5 Validation of AOD

In the offline mode, the LSA algorithm takes AOD as input and also updates AOD values in the process of optimization. Here we compare the updated AOD values from the optimization with AOD measurements at ten MODASRVN sites. Data for 500nm were transformed to 550nm in order to make the comparison with retrievals from MODIS observations. The monthly statistics of AOD were added into the surface albedo/reflectance retrieving procedure by constraining the physical range of optical depth. The overall comparison showed that the aerosol estimations have a small bias of 0.029 which means the algorithm can well capture the aerosol information in the TOA signals (Figure 4.10). However, there were some over- and under-estimations which deteriorated the RMSE to 0.1 when either the real AOD values fell out of the major monthly distribution or a large variation of AOD was found for that particular time period.

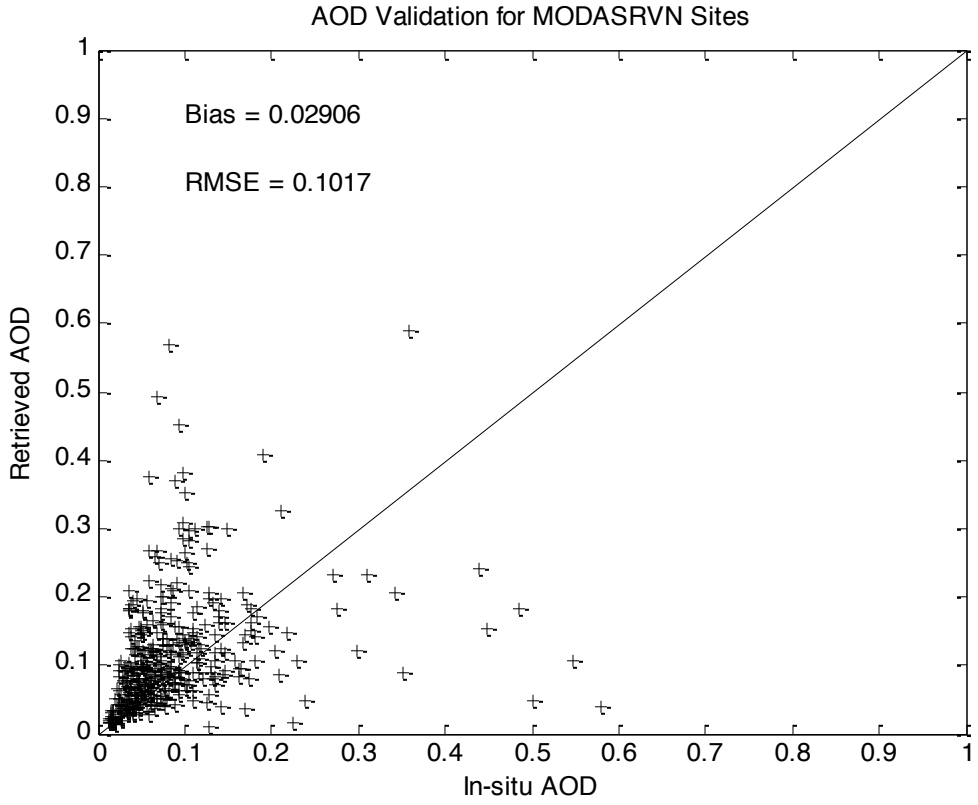


Figure 4.10. Validation summary of AOD at 550nm over MODASRVN sites for the year of 2005.

4.2.6 Summary of Accuracy and Precision S

The statistics of albedo validation is given in two groups: non-snow surfaces and snow surfaces. Non-snow surfaces use the data from SURFRAD sites and snow surfaces include sites of GC-Net. The performance of the LSA algorithm over both surfaces satisfies the requirements of F&PS in terms of precision and accuracy. The listed accuracy requirement for ABI albedo is 0.08 albedo unit and the error (RMSE) of our retrievals is 0.01 over both surfaces. In terms of RMSE (precision), we achieve a value of 0.06 over non-snow sites and 0.07 over snow sites while the F&PS requirement is 10% (Table 4.10). For the real ABI data, we expect even higher accuracy, since the ABI data have a temporal resolution of 15 minutes, providing sufficient data within a shorter time period, which is extremely important for the cases of rapidly changing surface properties, such as transitions between snowfall and snow melting.

Table 4.10. Summary of albedo validation results

	Non-snow	Snow	F&PS Requirement
Accuracy(Bias)	0.01	0.01	0.08
Precision(RMSE)	0.06	0.07	10%
R^2	0.82	0.84	N/A

Statistics of validation results of BRF over red and NIR bands are summarized in Table 4.11. The bias of our estimate is well below the requirement. In terms of RMSE (precision), our retrievals over all bands are also below 0.05.

Table 4.11. Summary of BRF validation results

	Red Band	NIR Band	Requirement
Accuracy(Bias)	0.008	0.003	0.08
Precision(RMSE)	0.027	0.047	5%
R^2	0.698	0.732	N/A

5 PRACTICAL CONSIDERATIONS

5.1 Numerical Computation Considerations

Accurate retrieval of albedo requires reliable acquisition of atmospheric parameters. Forward running of atmospheric radiatve transfer model is time-consuming and not suitable for operational retrieval of albedo. Instead, the LSA algorithm pre-runs the atmospheric radiative transfer at some given conditions and stores the parameters into the LUTs to save computational time.

The current version of the LSA algorithm includes an optimization process. To speed up the iterative process, we may have to limit the number of iterations or adjust the iteration convergence criteria.

5.2 Programming and Procedural Considerations

The LSA algorithm is purely a pixel-by-pixel algorithm. However, it requires a time series of clear-sky observations to achieve enough information to inverse BRDF models. Given the data volume of full disk albedo products, it is inefficient to gather a stack time series data over all pixels at each ABI scanning time. Given that the BRDF parameters do not vary greatly over a short period of time, we use the pre-calcualted BRDF parameters from the previous day to save computational time. In order to achieve this, we divide our algorithm into two parts, online and offline modes, respectively.

5.3 Quality Assessment and Diagnostics

The retrieval process of albedo will be monitored and the retrieval quality will be assessed. A set of quality flags and metadata will be generated with the albedo product for retrieval diagnostics. These flags will indicate the retrieval conditions, including the land/water mask, solar zenith angle and local zenith angle ranges. These flags also indicates the data quality (Is the data quality of AOD available? Is a routine BRDF retrieval algorithm successful? Which path is used to calculate LSA and BRF). The detailed information is documented in Section 3.5.

5.4 Exception Handling

The LSA algorithm checks for conditions where the albedo retrieval cannot be performed. These conditions include the failure of sensors, such as saturated channels or missing values. They also include the conditions when continuous clouds are present so that there are no enough clear-sky observations. However, the LSA algorithm tries to avoid using filling values if possible, in order to produce continuous and consistent

products. The LSA algorithm selects various paths to calculate albedo and BRF in the online mode. The filling value of BRF is used only when no BRDF parameter is retrieved and the current observation is cloudy. The LSA algorithm also considers the availability of model data. ABI ACM is required. The LSA algorithm cannot run without cloud mask. However, if there is no AOD available, the offline mode is still able to run normally and the online mode will use a default AOD value to calculate diffuse irradiance ratio and carry out atmospheric correction.

5.5 Algorithm Validation

A summary of our previous validation results has been given in Section 4. In order to quantify the retrieval errors and improve the inversion algorithm, we need to carry out more extensive validation work before and after launch of the GOES-R satellite. Albedo is continuously measured by several surface measurement networks, such as Atmospheric Radiation Measurement at the Southern Great Plains, SURFRAD, and Ameriflux projects. Albedo measurements at more than a hundred sites are available for pre-launch and post-launch validation. We have conducted albedo validation extensively during recent years (Chen et al. 2008; Liang et al. 2002; Liang et al. 2005b), and will continue this activity for the ABI albedo product over more surface types.

However, the spatial effects of validation LSA products have not been well addressed in current ATBD. The scale effect or the change of support is always a problem in validating satellite land products, especially those of moderate or coarse resolutions. Due to the practical challenges and the availability of measurements, in situ measurements of albedo from pyranometers are directly compared with satellite retrievals of albedo in our current validation activities and many other published investigations of albedo validation. We've noticed this problem. The scale effect has to be accounted in order to better evaluate accuracy and precision of GEOS-R albedo products. We have submitted a GOES-R cal/val proposal entitled "validating GOES-R land surface shortwave radiation products" to further investigate this issue.

6 ASSUMPTIONS AND LIMITATIONS

The following sections describe the assumptions in developing and estimating the performance of the current version of ABI LSA algorithm. The limitations and potential algorithm improvement are also discussed.

6.1 Performance

The following assumptions have been made in developing and estimating the performance of the ABI LSA algorithm:

- Surface BRF is modeled by the revised linear kernel function with three coefficients.
- Surface anisotropy is constant within days through a moving window and can be represented by the linear kernel model.
- The reciprocity principle is valid at ABI resolutions.

6.2 Assumed Sensor Performance

The ABI LSA algorithm requires a time series of clear sky TOA reflectance inputs. The number of clear sky observations within a short time period will influence the retrieval quality of LSA and corresponding land surface reflectance by-products. Additionally, the algorithm relies on the cloud mask product to distinguish clear-sky observations from cloud sky observations. The retrieval accuracy also depends on the quality of cloud mask.

6.3 Algorithm Improvement

The introduction of prior knowledge such as the aerosol types, BRDF models and albedo climatologies will improve the retrieval quality of LSA and land surface reflectance. Currently, we use the multiyear's mean and variance of MODIS albedo products as one of the constraints in our optimization code. We are currently working on analyzing more existing satellite albedo/BRDF products and in an effort to incorporate as much background knowledge as possible.

7 REFERENCES

Chen, Y.M., Liang, S., Wang, J., Kim, H.Y., & Martonchik, J.V. (2008). Validation of MISR land surface broadband albedo. *International Journal of Remote Sensing, 29*, 6971-6983

Duan, Q.Y., Gupta, V.K., & Sorooshian, S. (1993). Shuffled complex evolution approach for effective and efficient global minimization. *Journal of Optimization Theory and Applications, 76*, 501-521

Duan, Q.Y., Sorooshian, S., & Gupta, V. (1992). Effective and efficient global opitimization for conceptual rainfall-runoff models. *Water Resources Research, 28*, 1015-1031

Govaerts, Y.M., Wagner, S., Lattanzio, A., & Watts, P. (2010). Joint retrieval of surface reflectance and aerosol optical depth from MSG/SEVIRI observations with an optimal estimation approach: 1. Theory. *Journal of Geophysical Research-Atmospheres, 115*

Liang, S., Yu, Y., & Defelice, T.P. (2005a). VIIRS narrowband to broadband land surface albedo conversion: formula and validation. *International Journal of Remote Sensing, 26*, 1019-1025

Liang, S.L. (2001). Narrowband to broadband conversions of land surface albedo I Algorithms. *Remote Sensing of Environment, 76*, 213-238

Liang, S.L. (2003). A direct algorithm for estimating land surface broadband albedos from MODIS imagery. *Ieee Transactions on Geoscience and Remote Sensing, 41*, 136-145

Liang, S.L. (2004). *Quantitative remote sensing of land surfaces*. Hoboken, New Jersey: John Wiley & Sons, Inc

Liang, S.L., Fang, H.L., Chen, M.Z., Shuey, C.J., Walthall, C., Daughtry, C., Morisette, J., Schaaf, C., & Strahler, A. (2002). Validating MODIS land surface reflectance and albedo products: methods and preliminary results. *Remote Sensing of Environment, 83*, 149-162

Liang, S.L., Shuey, C.J., Russ, A.L., Fang, H.L., Chen, M.Z., Walthall, C.L., Daughtry, C.S.T., & Hunt, R. (2003). Narrowband to broadband conversions of land surface albedo: II. Validation. *Remote Sensing of Environment, 84*, 25-41

Liang, S.L., Strahler, A.H., & Walthall, C. (1999). Retrieval of land surface albedo from satellite observations: A simulation study. *Journal of Applied Meteorology, 38*, 712-725

Liang, S.L., Stroeve, J., & Box, J.E. (2005b). Mapping daily snow/ice shortwave broadband albedo from Moderate Resolution Imaging Spectroradiometer (MODIS): The improved direct retrieval algorithm and validation with Greenland in situ measurement. *Journal of Geophysical Research-Atmospheres, 110*

Maignan, F., Breon, F.M., & Lacaze, R. (2004). Bidirectional reflectance of Earth targets: Evaluation of analytical models using a large set of spaceborne measurements with emphasis on the Hot Spot. *Remote Sensing of Environment, 90*, 210-220

NOAA (2009). GOES-R Series Ground Segment Project Functional and Performance Specification. In

Pinty, B., Roveda, F., Verstraete, M.M., Gobron, N., Govaerts, Y., Martonchik, J.V., Diner, D.J., & Kahn, R.A. (2000a). Surface albedo retrieval from Meteosat - 1. Theory. *Journal of Geophysical Research-Atmospheres, 105*, 18099-18112

Pinty, B., Roveda, F., Verstraete, M.M., Gobron, N., Govaerts, Y., Martonchik, J.V., Diner, D.J., & Kahn, R.A. (2000b). Surface albedo retrieval from Meteosat - 2. Applications. *Journal of Geophysical Research-Atmospheres, 105*, 18113-18134

Qin, W.H., Herman, J.R., & Ahmad, Z. (2001). A fast, accurate algorithm to account for non-Lambertian surface effects on TOA radiance. *Journal of Geophysical Research-Atmospheres, 106*, 22671-22684

Schaaf, C., Martonchik, J., Pinty, B., Govaerts, Y., Gao, F., Lattanzio, A., Liu, J., Strahler, A., & Taberner, M. (2008). Retrieval of Surface Albedo from Satellite Sensors. In S. Liang (Ed.), *Advances in Land Remote Sensing: System, Modeling, Inversion and Application* (pp. 219-243). New York: Springer

Schaaf, C.B., Gao, F., Strahler, A.H., Lucht, W., Li, X.W., Tsang, T., Strugnell, N.C., Zhang, X.Y., Jin, Y.F., Muller, J.P., Lewis, P., Barnsley, M., Hobson, P., Disney, M., Roberts, G., Dunderdale, M., Doll, C., d'Entremont, R.P., Hu, B.X., Liang, S.L., Privette, J.L., & Roy, D. (2002). First operational BRDF, albedo nadir reflectance products from MODIS. *Remote Sensing of Environment, 83*, 135-148

Wagner, S.C., Govaerts, Y.M., & Lattanzio, A. (2010). Joint retrieval of surface reflectance and aerosol optical depth from MSG/SEVIRI observations with an optimal estimation approach: 2. Implementation and evaluation. *Journal of Geophysical Research-Atmospheres, 115*

Wang, Y.J., Lyapustin, A.I., Privette, J.L., Morisette, J.T., & Holben, B. (2009). Atmospheric Correction at AERONET Locations: A New Science and Validation Data Set. *Ieee Transactions on Geoscience and Remote Sensing, 47*, 2450-2466

Yang, F.L., Mitchell, K., Hou, Y.T., Dai, Y.J., Zeng, X.B., Wang, Z., & Liang, X.Z. (2008). Dependence of Land Surface Albedo on Solar Zenith Angle: Observations and Model Parameterization. *Journal of Applied Meteorology and Climatology, 47*, 2963-2982

APPENDIX 5: COMMON ANCILLARY DATA SETS FOR SURFACE ALBEDO AND SURFACE REFLECTANCE

(Note: All Tables listed in this Appendix refer to the Tables in AIADD)

1. Ancillary Data Sets

1.1 LAND_MASK_NASA_1KM

a. Data description.

Description: The land/ocean mask is derived from the NASA EOS project supplied static dataset as well as World Vector Shoreline data and DTED DEM data provided by NIMA (then DMA) and bathymetric data provided by the oceanographic community. This is further described here: http://www-modis.bu.edu/brdf/lw.html
The original global binary file, version 3, produced in 2003 by Robert Wolfe, was converted to netCDF and HDF for usage in the framework. The original binary file is available at ftp://landsc1.nascom.nasa.gov/pub/outgoing/lwm_v03m/
Resolution: The land/ocean mask is stored in a 1 km geographic (geodetic) projection.
Filename: lw_geo_2001001_v03m.nc
Origin: Created by SSEC/CIMSS based on NASA MODIS collection 5
Size: 890 MB.
Static/Dynamic: Static
Values:

0 = Shallow ocean
1 = Land (Nothing else but land)
2 = Ocean coastlines and lake shorelines
3 = Shallow inland water
4 = Ephemeral water
5 = Deep inland water
6 = Moderate or continental ocean
7 = Deep ocean

b. Interface in the framework

SUBROUTINE **Read_LandMask_Interface(Ctxt, Return_Status)**

Where,

Ctxt – context of the algorithm, in which input and output variables are contained.
Data Type: Structure (**Land_Mask_NASA_1KM_Ctxt)**
Data Size: Static
--- See Table 2.5 for details

Return_Status – return status of the routine.
Data Type: Integer (long)
Data size: Static

c. Context: Variables required by the interface

Variable Name: Ctxt%SegmentInfo%**Current_Column_Size**
Data Type: **Integer(long)**
Data Size: **Static**

Variable Name: Ctxt%SegmentInfo%**Current_Row_Size**
Data Type: **Integer(long)**
Date Size: **Static**

Variable Name: Ctxt%SATELLITE_DATA_Src1_T00%Data%Nav%**Lat**
DataType: **Real(single)**
Data Size: **Dynamic**

Variable Name: Ctxt%SATELLITE_DATA_Src1_T00%Data%Nav%**Lon**
Data Type: **Real(single)**
Data Size: **Dynamic**

Variable Name: Ctxt%SATELLITE_DATA_Src1_T00%Data%Nav%**SpaceMsk**
Data Type: **Integer(byte)**
Data Size: **Dynamic**

For details about each variables of each data structure:

Ctxt – See **Table 2.5**
Ctxt%SegmentInfo – See **Table 1.1**
Ctxt%SATELLITE_DATA_Src1_T00 – See **Table 1.2**
Ctxt%SATELLITE_DATA_Src1_T00%Data – See **Table 1.6**
Ctxt%SATELLITE_DATA_Src1_T00%Data%Nav – See **Table 1.7**

d. Context: Variables output by the interface.

Variable Name: Ctxt%LAND_MASK_Src1_T00%**Mask**
Data Type: **Integer(byte)**

Data Size: **Dynamic**

Variable Name: Ctxt% LAND_MASK_Src1_T00%**SDSName**
Data Type: **Character(LEN=32)**
Data Size: **Static**

Variable Name: Ctxt% LAND_MASK_Src1_T00%**Filename**
Data Type: **Character(LEN=128)**
Data Size: **Static**

Variable Name: Ctxt% LAND_MASK_Src1_T00%**Directory**
Data Type: **Character(LEN=128)**
Data Size: **Static**

For details about data type and size of each variables of each data structure:

Ctxt – See **Table 2.5**
Ctxt% LAND_MASK_Src1_T00 – See **Table** 1.15

e. Interpolation description

The closest point is used for each satellite pixel:

Subroutine Read_Mask_Byte is called by the interface Read_LandMask_Interface. And in Read_Mask_Byte, Subroutine read_land_sfc_netcdf is called to read a given land surface netcdf file and to find the closest point for each satellite pixel. There is no curvature effect. Before being read in by the various algorithms, if the current pixel is a space pixel, as determined by the space pixel mask, the land mask given to the algorithm is set to the missing value sentinel.

Closest point mapping algorithm:

Input Ancillary data is in a regular grid.
Input:
Navigation information of target point(satellite pixel):
Lat / Lon (Latitude / Longitude)
Ancillary data grid information:
sLat / sLon: Grid start Latitude / Longitude
dLat / dLon: Grid increase step of Latitude / Longitude
Output:
The closest ancillary data grid index: iLat / iLon
iLat = (Lat – sLat) / dLat + 1
iLon = (Lon – sLon) / dLon + 1

1. From satellite observation Latitude / Longitude, define a block area (rectangular: northwest, northeast, southwest, southeast) in which the satellite pixel is included.
2. Read in the sub grid of ancillary grid covering the block area.
3. In Latitude / Longitude space, use the ancillary data closest to the satellite pixel.