AVHRR Polar Pathfinder Twice-Daily 5 km EASE-Grid Composites, Version 3

USER GUIDE

How to Cite These Data

As a condition of using these data, you must include a citation:

Fowler, Chuck, James Maslanik, Terry Haran, Ted Scambos, Jeffrey Key, and William Emery. 2000, updated 2007. *AVHRR Polar Pathfinder Twice-daily 5 km EASE-Grid Composites V003* [list dates of data used]. Boulder, Colorado USA: National Snow and Ice Data Center. Digital media. https://doi.org/10.5067/HRMXN6PE1Q0Q. [Date Accessed].

FOR QUESTIONS ABOUT THESE DATA, CONTACT NSIDC@NSIDC.ORG

FOR CURRENT INFORMATION, VISIT https://nsidc.org/data/NSIDC-0066

NSIDC National Snow and Ice Data Center

TABLE OF CONTENTS

1 DATA DESCRIPTION

1.1 Parameters

The AVHRR Polar Pathfinder twice-daily 5 km EASE-grid composites contain the following parameters:

- Channel 1 Top of the Atmosphere (TOA) reflectance
- Channel 2 TOA Reflectance
- Channel 3 TOA Brightness Temperature
- Channel 3A TOA Reflectance
- Channel 3B TOA Brightness Temperature
- Channel 4 TOA Brightness Temperature
- Channel 5 TOA Brightness Temperature
- Clear Sky Surface Broadband Albedo
- Clear Sky Surface Skin Temperature
- Solar Zenith Angle
- Satellite Elevation Angle
- Sun-satellite Relative Azimuth Angle
- Surface Type Mask
- Cloud Mask
- Universal Coordinated Time (UTC) of Acquisition

Table 1. AVHRR 5 km Data Products and Volumes for Both Hemispheres

Parameter	Bytes/Pixel	Times/Day	MB/Day	Units	Scaling Factor	Data Range
Channel 1 TOA Reflectance	2	2	23	% reflectance	0.1	0 to 1000
Channel 2 TOA Reflectance	2	2	23	% reflectance	0.1	0 to 1000
Channel 3 TOA Brightness Temperature	2	2	23	kelvin	0.1	<1900 to 3100>
Channel 3A TOA Reflectance	2	2	23	% reflectance	0.1	<0 to 1000>
Channel 3B TOA Brightness Temperature	2	2	23	kelvin	0.1	<1900 to 3100>
Channel 4 TOA Brightness Temperature	2	2	23	kelvin	0.1	<1900 to 3100>
Channel 5 TOA Brightness Temperature	2	2	23	kelvin	0.1	<1900 to 3100>

Parameter	Bytes/Pixel	Times/Day	MB/Day	Units	Scaling Factor	Data Range
Clear Sky Surface Broadband Albedo	2	2	23	% albedo	0.1	<0 to 1000>
Clear Sky Surface Skin Temperature	2	2	23	kelvin	0.1	<1900 to 3100>
Solar Zenith Angle	2	2	23	degrees	0.1	<0 to 900>
Satellite Elevation Angle	2	2	23	degrees	0.1	0 to 900
Sun-satellite Relative Azimuth Angle	2	2	23	degrees	0.1	0 to 1800
Surface Type Mask	1	1	6	n/a	n/a	10 to 60
Cloud Mask	1	2	12	n/a	n/a	0 to 7, 128
Time of Acquisition (UTC)	1	2	12	hour	0.1	0 to 239[a] 0 to 244[b]
TOTAL	**260 MB/Day**					

[a] Versions 1 and 2 data

[b] Version 3 data

Data were scaled (except for surface type and cloud mask, indicated by N/A) with a scaling factor. The original data values can be recovered using the following equation: orig_value=Scaling_Factor * scaled_value.

1.2 Data Access

Data are currently unavailable because of data errors.

We recommend users access the following NOAA Climate Data Record (CDR) which contains the AVHRR Polar Pathfinder (APP) product: see, NSIDC-0745. If you need information pertaining to the data errors, please see Quality, Errors, and Limitations of the guide documentation. If you have further questions, contact NSIDC User Services by e-mail: nsidc@nsidc.org or by telephone +1 303.492.6199.

1.3 File Information

1.3.1 Format

Individual parameters are stored as 1-byte or 2-byte raster files for each date, time, and hemisphere. Cell size is 5 km, with either 1 or 2 bytes per cell. Refer to the Naming Convention section of this document. Byte order is big-endian. The Northern Hemisphere grid is 1805 pixels wide by 1805 pixels high, centered on the North Pole. The Southern Hemisphere grid is 1605 pixels wide by 1605 pixels high, centered on the South Pole.

Formats for 1-byte raster files are summarized in

Table 3, Table 4, and Table 4. Formats for 2-byte raster files are summarized in the Parameters section.

Table 2. Surface Type Mask (SMSK) Cell Values

Cell Value	Description
10	open water
20-29	mostly first-year ice (range indicates total ice concentration in tens of percent; for example, a value of 20 indicates more first-year ice than multiyear ice with a total ice concentration between one and ten percent)
30-39	mostly multi-year ice (range is same as above; however, multiyear ice is dominant)
40	bare land
50	snow covered land
60	ice sheet

Table 3. Cloud Mask (CMSK) Bit Number Values Versions 1 and 2 Data (1981-1993)

Bit Number	Description	Bit Value
Bit 0	The least significant bit. Cloud mask from Cloud and Surface Parameter Retrieval (CASPR) single-day algorithm.	0 - clear
		1 - cloud
Bit 1	Cloud mask based on a time-filtered series of channel four.	0 - clear
		1 - cloud
Bit 2	Cloud mask based on multi day CASPR algorithm.	0 - clear
		1 - cloud
Bit 7	Set if no valid data are present.	0 - valid data

		1 - missing data

Table 4. Cloud Mask (CMSK) Bit Number Values Versions Version 3 Data (1994-2005)

Bit Number	Description	Bit Value
Bit 0	The least significant bit. Cloud mask from CASPR single-day algorithm.	0 - clear
		1 - cloud
Bit 1	Cloud mask from updated CASPR multi day algorithm.	0 - clear
		1 - cloud
Bit 2	Set if no valid data are present.	0 - valid data
		1 - missing data

Time Mask Cell Values

Data are in increments of 0.1 hours with cell values ranging from 0 to 239 for Version 1 and 2 data, and 0 to 244 for Version 3 data.

1.3.2 File Contents

File Sizes:

- Northern Hemisphere: 6.52 MB (2-byte) and 3.26 MB (1-byte) per granule.
- Southern Hemisphere: 5.15 MB (2-byte) and 2.58 MB (1-byte) per granule.

1.3.3 Naming Convention

This section explains the file naming convention used for this product with an example.

Example file name: `a16_n005_2005181_9999_smsk.v3`

```
app_h005_yyyyddd_tttt_zzzz.xx
```

Table 5. File Name Variables and Descriptions

Value	Description
a	indicates AVHRR Polar Pathfinder
pp	represents which satellite the data came from such as NOAA-16
h	represents the Northern or Southern hemisphere (n for northern and s for southern)
005	represents the 5 km resolution
yyyy	4-digit year

Value	Description
ddd	3-digit day of year
tttt	the hour (0400 or 1400 for north files, 0200 or 1400 for south files, and 9999 for north and south surface type mask files - one per day)
zzzz	the file type as listed below: 2-byte raster files: albd - clear sky surface broadband albedo chn1 - AVHRR channel 1 chn2 - AVHRR channel 2 chn3 - AVHRR channel 3 chn4 - AVHRR channel 4 chn5 - AVHRR channel 5 sael - satellite elevation angle solz - solar zenith angle reaz - sun-satellite relative azimuth angle temp - clear sky surface skin temperature 1-byte raster files: smsk - surface type mask cmsk - cloud mask time - time of acquisition (UTC) **GISMO File Names:** The file naming convention is the same for AVHRR files ordered through the GISMO interface except that the text `subset_` is appended to the front of the file name.
xx	represents the version number of the data, such as v3 for Version 3

1.4 Spatial Information

1.4.1 Coverage

Spatial coverage extends from 48.4 degrees to 90.0 degrees north latitude, and from 53.2 degrees to 90.0 degrees south latitude. The actual coverage extends beyond these limits in the grid corners. Table 6 and Table 7 summarize the values of corner pixels for each hemisphere.

Note: For the Northern Hemisphere, the center of the tangent pixels is 48.42649 north latitude and the outer edge is 48.40237 north latitude. For the Southern Hemisphere, the center of the tangent pixels is 53.21244 south latitude and the outer edge is 53.18868 south latitude.

Table 6. Northern Hemisphere

Corner	**Center of Corner Pixel**	**Outer Edge of Corner Pixel**
Upper left	29.74956° N, 135.00000° W	29.71269° N, 135.00000° W
Upper right	29.74956 N°, 135.00000° E	29.71269° N, 135.00000° E
Lower left	29.74956 N°, 45.00000° W	29.71269° N, 45.00000° W
Lower right	29.74956 N°, 45.00000° E	29.71269° N, 45.00000° E

Table 7. Southern Hemisphere

Corner	**Center of Corner Pixel**	**Outer Edge of Corner Pixel**
Lower left	36.99339° S, 135.00000° W	36.95776° S, 135.00000° W
Lower right	36.99339° S, 135.00000° E	36.95776° S, 135.00000° E
Upper left	36.99339° S, 45.00000° W	36.95776° S, 45.00000° W
Upper right	36.99339° S, 45.00000° E	36.95776° S, 45.00000° E

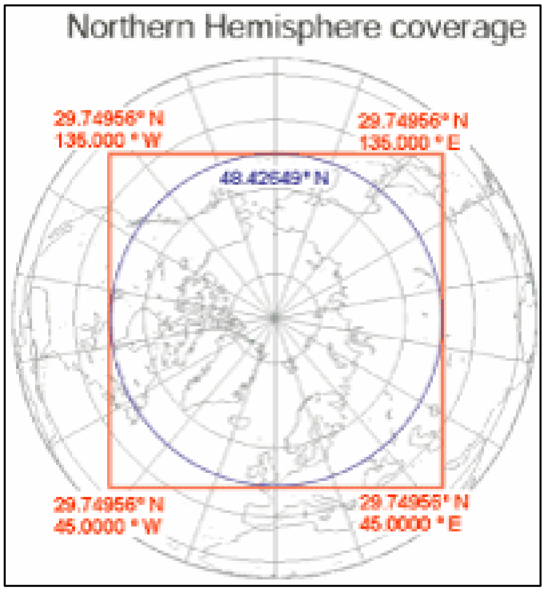

Figure 1. Coordinates shown for the outer edges of corner pixels. The latitude in blue indicates the center of tangent pixels.

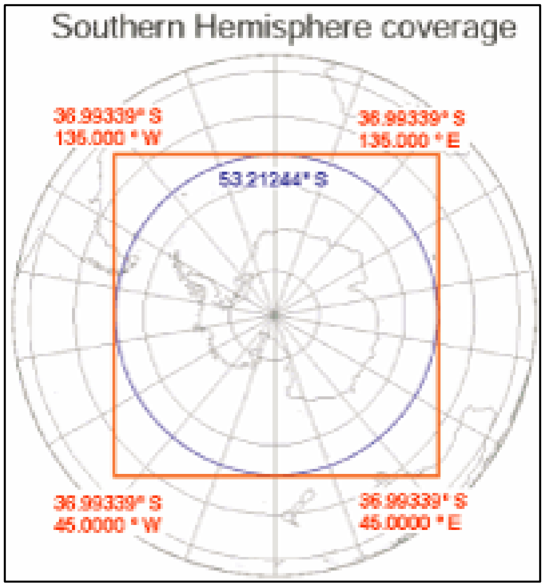

Figure 2. Coordinates shown for the outer edges of corner pixels. The latitude in blue indicates the center of tangent pixels.

1.4.2 Resolution

AVHRR Global Area of Coverage (GAC) data are generated from the original Local Area Coverage (LAC) imagery by applying a sub sampling procedure. For every third line in the image, a series of five pixels is selected. The first four pixels are averaged, and the fifth pixel is removed. Data are regridded to the EASE-Grid projection with a 5 km pixel spacing.

1.4.3 Swath Description

The data are received as swath data. The AVHRR instrument scans in the across-track direction with a continuously rotating scan mirror, viewing a swath of over 100 degrees and up to 55 degrees off-nadir.

1.4.4 Geolocation

1.4.4.1 Projection

Data are georeferenced to the Equal Area Scalable Earth-Grid (EASE-Grid) projection, an azimuthal equal area projection. Please review the EASE-Grid web site for general information on the EASE-Grid, and the Summary of NOAA/NASA Polar Pathfinder Grid Relationships web page for details of map projection parameters.

1.4.4.2 Grid Description

The data are provided in EASE-grid projection. The Northern Hemisphere grid is 1805 pixels by 1805 pixels, centered on the North Pole. The Southern Hemisphere grid is 1605 pixels by 1605 pixels, centered on the South Pole. Cell size is 5 km. Grid coordinates begin in the upper left corner of the grid. See the NOAA/NASA Polar Pathfinder web page for more information.

1.5 Temporal Information

1.5.1 Coverage

Temporal coverage is from 24 July 1981 to 30 June 2005. Exclusive of missing dates, the average availability of data at any location is greater than 95 percent for the north polar regions and 96 percent for the south polar regions. Table 8, Table 9, and Table 10 summarize the temporal coverage for Versions 1, 2, and 3 data. Table 11Table 11 summarizes the missing dates of temporal coverage.

Table 8. Temporal Coverage for Version 1 Data

Northern Hemisphere	
1982	All
1983	11 February - 20 November
1986	1 January - 2 June 15 June - 21 June 7 July - 31 December
1987	All
1988	All
1989	1 January - 30 June 21 July - 31 December
1990	1 May - 31 December
Southern Hemisphere	
1982	All
1983	11 February - 31 January
1984	1 January - 20 November
1986	1 January - 2 June 13 June - 21 June 7 July - 31 December
1987	All
1988	All
1989	All

Table 9. Temporal Coverage for Version 2 Data

Northern Hemisphere	
1981	24 July - 31 December
1983	1 January - 10 February 21 November - 31 December
1984	All
1985	All
1986	3 June - 12 June 22 June - 16 July
1989	1 July - 20 July
1990	1 January - 30 April
1991-1993	All
Southern Hemisphere	
1981	24 July - 31 December
1982	All
1983	1 January - 10 February
1984	21 November - 31 December
1985	All
1986	3 June - 14 June 22 June - 6 July
1987-1993	All

Table 10. Temporal Coverage for Version 3 Data

Northern and Southern Hemispheres	
1994	1 January - 12 September
1995	19 January - 31 December
1996	1 January - 31 December
1997	1 January - 31 December
1998	1 January - 31 December
1999	1 January - 31 December
2000	1 January - 31 December
2001	2 January - 31 December
2002	1 January - 31 December
2003	1 January - 31 December
2004	1 January - 31 December
2005	1 January - 30 June

Table 11. Missing Dates of Temporal Coverage

Missing Dates of Temporal Coverage	
Year	Dates
1981	22 August - 31 August 29, 30 November
1982	2 June 11 August 25, 26, 29 September 26 November 28 - 31 December
1983	2 - 7, 9 - 13 January 1 - 3 May 6, 17 - 23 August 21 - 26 September 12, 13, 19, 20, 23 - 26, 28 - 31 December
1984	1 - 10, 14, 15 January 18 - 22, 24, 25 February 23 - 30 March 10, 14 - 19 April 04 May 4 - 6 June 13, 18 November 6, 9 December
1985	1 - 3, 13, 17, 20, 23 January 8 - 10 February 16 March 11, 12 April 20 November 7, 14 December
1986	28 - 31 January 14, 15 March 18 - 21 May 13, 14 June 4 November 27 - 31 December
1987	October 31
1988	25 February 25 July 3, 24 November
1989	18 - 21 May
1990	18 - 21 February
1991	17 - 27 July

Missing Dates of Temporal Coverage	
1992	1 January 23 May 15 September
1994	14 September - 31 December
1995	1 - 20 January 5, 19 - 21 March 31 May 13 October
1996	31 March 21, 22 May
1997	8, 17, 18 February 12 March - 22 April 11, 14, 23 June
1998	23 January 10 - 12 February 28, 29, 31 May 20 June 7 - 9, 13, 16 - 19 July, 4 - 9 August
1999	Complete
2000	8 August
2001	1 January
2002	Complete
2003	Complete
2004	Complete
2005	Complete

1.5.2 Resolution

Twice-daily composites are produced at target times of 0400 (morning) and 1400 (afternoon) local time for the Northern Hemisphere, and 0200 (morning) and 1400 (afternoon) local time for the Southern Hemisphere. The precise local time for 90 percent of the cells in each grid is within one hour of the target time. See the Data Acquisition and Processing section of this document for more information.

2 DATA ACQUISITION AND PROCESSING

2.1 Background

2.1.1 Optical Remote Sensing

Remote sensing in the optical part of the electromagnetic spectrum (400-2500 nm) involves measuring the combined reflectance of solar radiation from both the Earth's surface and atmosphere. There are several specific definitions of reflectance referenced in Table 12, but in general, reflectance involves the diffuse scattering of light by a geometrically complex surface (Hapke 1993). Reflectance varies according to the degree of collimation, the incident irradiance, and the collimation of the detector. Collimation refers to the degree of angular diffusion of the incident light or the size of the angular field of view of the detector. For instance, direct beam solar energy is considered highly collimated whereas the diffuse sky radiance is uncollimated; a narrow-angle field of view detector can be considered collimated while a hemispherical sensor such as a pyranometer is an uncollimated detector (Diner et al. 1999).

Table 12. Commonly-used Terms for Reflectance

Name	Definition
Bidirectional Reflectance Distribution Function (BRDF)	Surface-leaving radiance divided by incident irradiance from a single direction
Bidirectional Reflectance Factor (BRF)	Surface-leaving radiance divided by radiance from a Lambertian reflector illuminated from a single direction
Hemispherical-Directional Reflectance Factor (HDRF)	Surface-leaving radiance divided by radiance from a Lamberian reflector illuminated under the same ambient conditions
Directional Hemispherical Reflectance (DHR)	Radiant exitance divided by irradiance under illumination from a single direction
Bihemispherical Reflectance (BHR)	Radiant exitance divided by irradiance under ambient illumination conditions

The percentage of incident solar radiation reflected back by an object is called albedo. Surface objects with a high albedo reflect more solar energy and appear as bright objects in a visible/near-infrared image. Objects with a low albedo reflect only a small portion of incident solar radiation and appear as dark objects on a visible/near-infrared image. Figure 3 summarizes how variations of reflectance depend upon the degree of collimation (Hapke 1993).

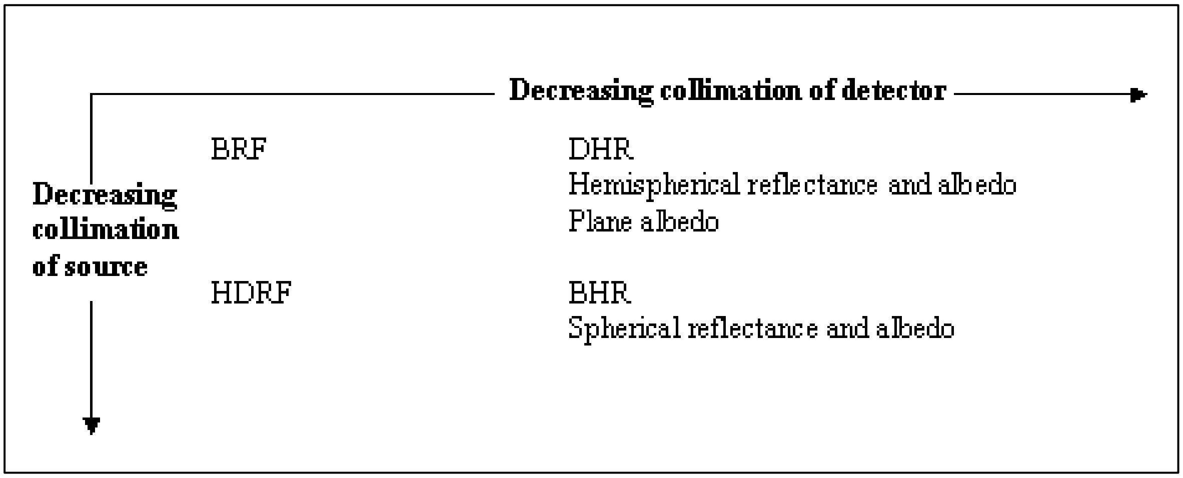
Decreasing collimation of detector
Decreasing collimation of source
BRF
DHR
Hemispherical reflectance and albedo
Plane albedo
HDRF
BHR
Spherical reflectance and albedo

Figure 3. Variations of Reflectance

AVHRR Channels 1 and 2 measure the upwelling radiance at the Top-of-Atmosphere (TOA) emanating from both the Earth surface and atmospheric scattering. TOA radiance is converted to a TOA BRF using the exo-atmospheric solar irradiance, Earth-Sun distance, solar zenith angle, and spectral response functions of the AVHRR channels. The surface albedo is obtained from the TOA BRF after applying an atmospheric correction and a conversion from directional reflectance to hemispherical reflectance. Figure 4 shows a summary of the solar and satellite angles used in processing the AVHRR Polar Pathfinder products.

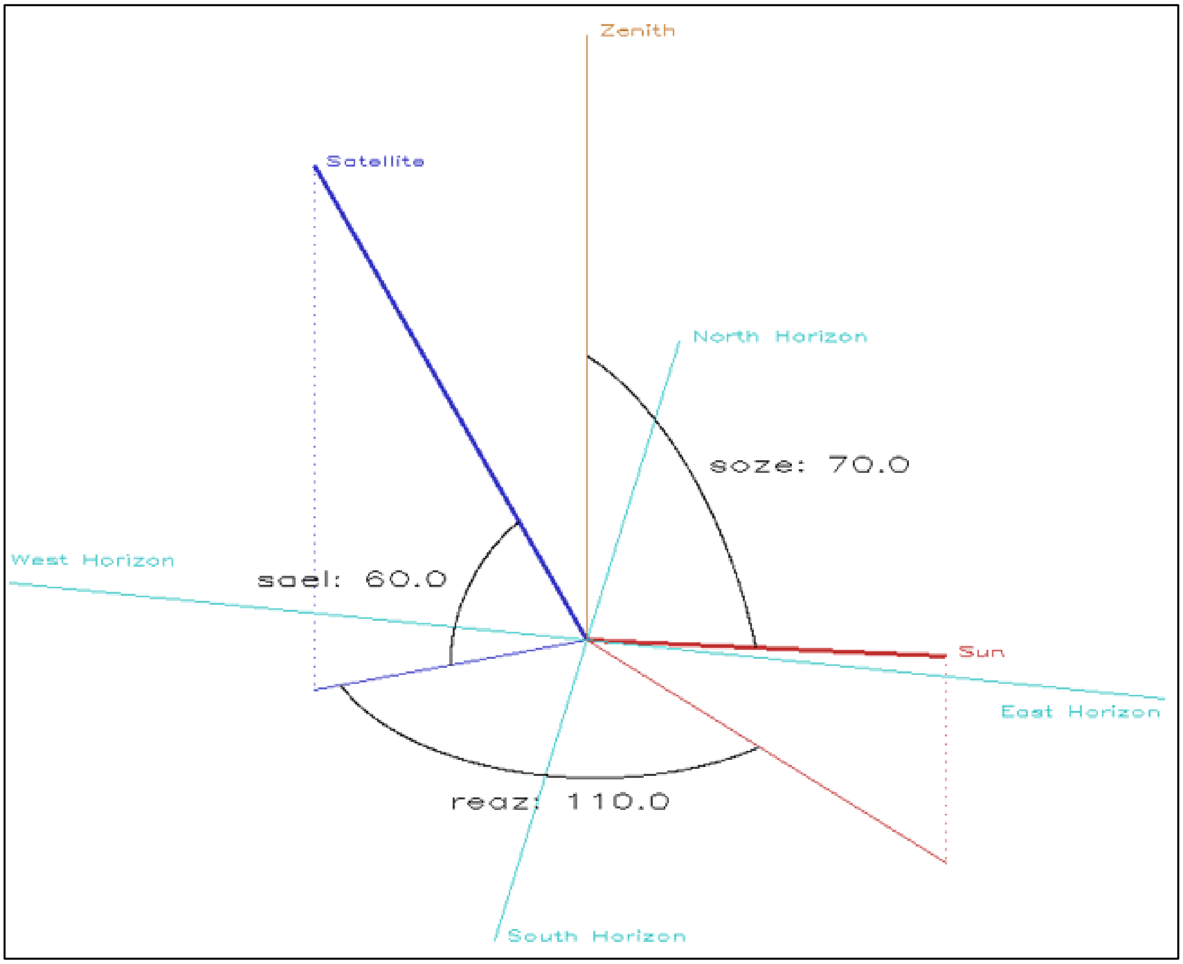

Figure 4. Summary of Solar and Satellite Angles Used in Processing AVHRR Polar Pathfinder Products

Channel 3A on the NOAA 16 AVHRR/3 instrument allows improved discrimination between snow and clouds by using the 1.6 µm wavelength. At 1.6 µm, snow has very low reflectance, while the reflectance of clouds remains high. Refer to Figure 5. Therefore, both cirrus and optically thick clouds can be directly classified and distinguished from snow at the 1.6 µm wavelength (Warren 1982).

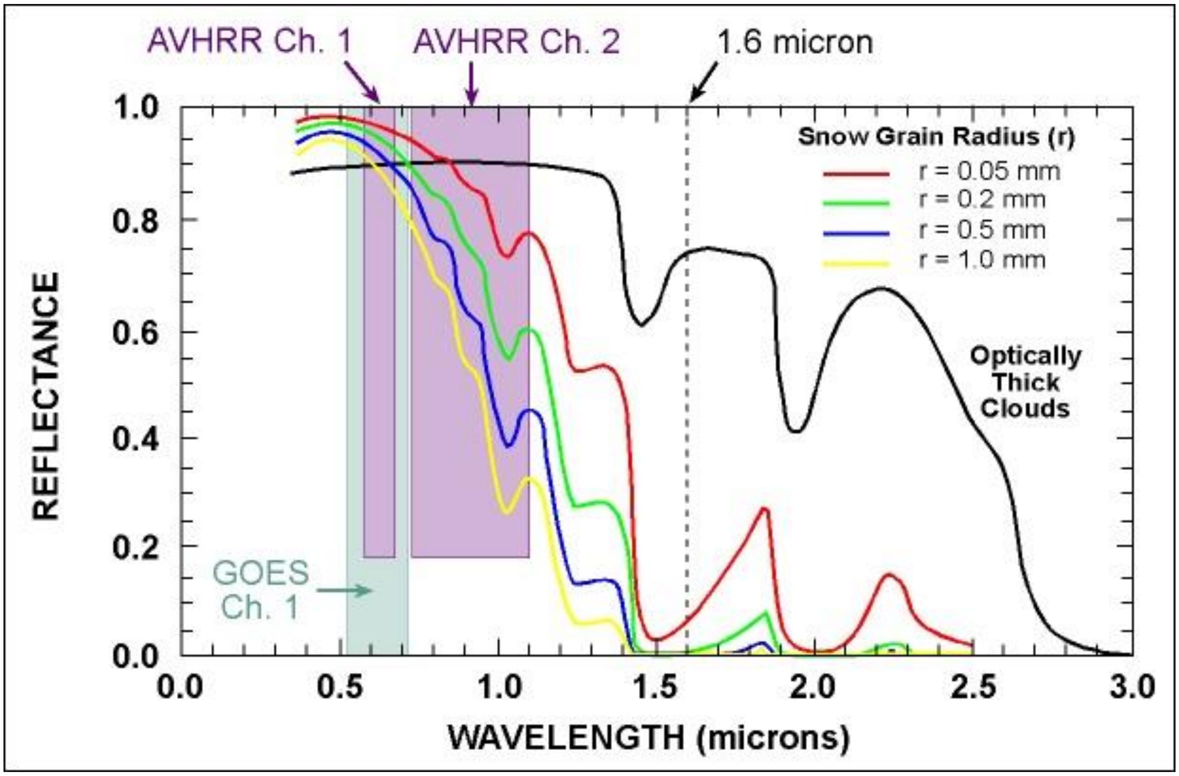

Figure 5. Satellite Channel Wavelengths in Micrometers (µm), and Typical Reflectance Spectra for Snow and Clouds

2.1.2 Thermal Remote Sensing

All objects radiate energy according to their blackbody temperature. A blackbody is a hypothetical object that absorbs all incoming thermal energy, but with none of that energy reflected or transmitted. Since no object can ideally absorb 100 percent of incident energy, a blackbody serves as a comparative measure of thermal emission. As the temperature of an object increases, the total amount of the emitted energy also increases, and the wavelength of that energy becomes shorter, as Wien's displacement law describes. According to the Stefan-Boltzman law, objects with a higher temperature give off more thermal energy per unit area than objects with a lower temperature. Remote sensing in the thermal infrared spectral region (approximately 7 µm to 14 µm) involves measuring the radiance of objects. More specifically, thermal remote sensing measures the differences in the ability of objects to absorb shortwave energy and emit it back as longwave energy. Researchers are primarily interested in measuring the actual temperature of objects, rather than radiance. Radiance is simply a measure of the emitted energy of an object, while temperature is a measure of the kinetic (thermal) energy of an object. AVHRR channels 3-5 measure TOA brightness temperature. Refer to the NOAA Polar Orbiter Data User's Guide for details on how radiance values are converted to surface temperature, and the Derivation Techniques and Algorithms section of this document for algorithms used in the AVHRR Polar Pathfinder Twice-Daily 5 km EASE-Grid Composites.

2.2 Acquisition

Refer to Figure 6 for a summary of the steps used in processing the AVHRR Level 1B GAC data, which were obtained from a variety of sources. Data from NCAR were used for initial development and testing of processing and algorithms. The GSFC DAAC supplied several years' worth of GAC data on 8 mm tapes. NSIDC copied GAC data from optical platters, loaned from JPL. A Digital Equipment Corporation (DEC) computer and VAX/VMS operating system were used for the copying. Finally, data from NOAA SAA were used to complete the GAC data set.

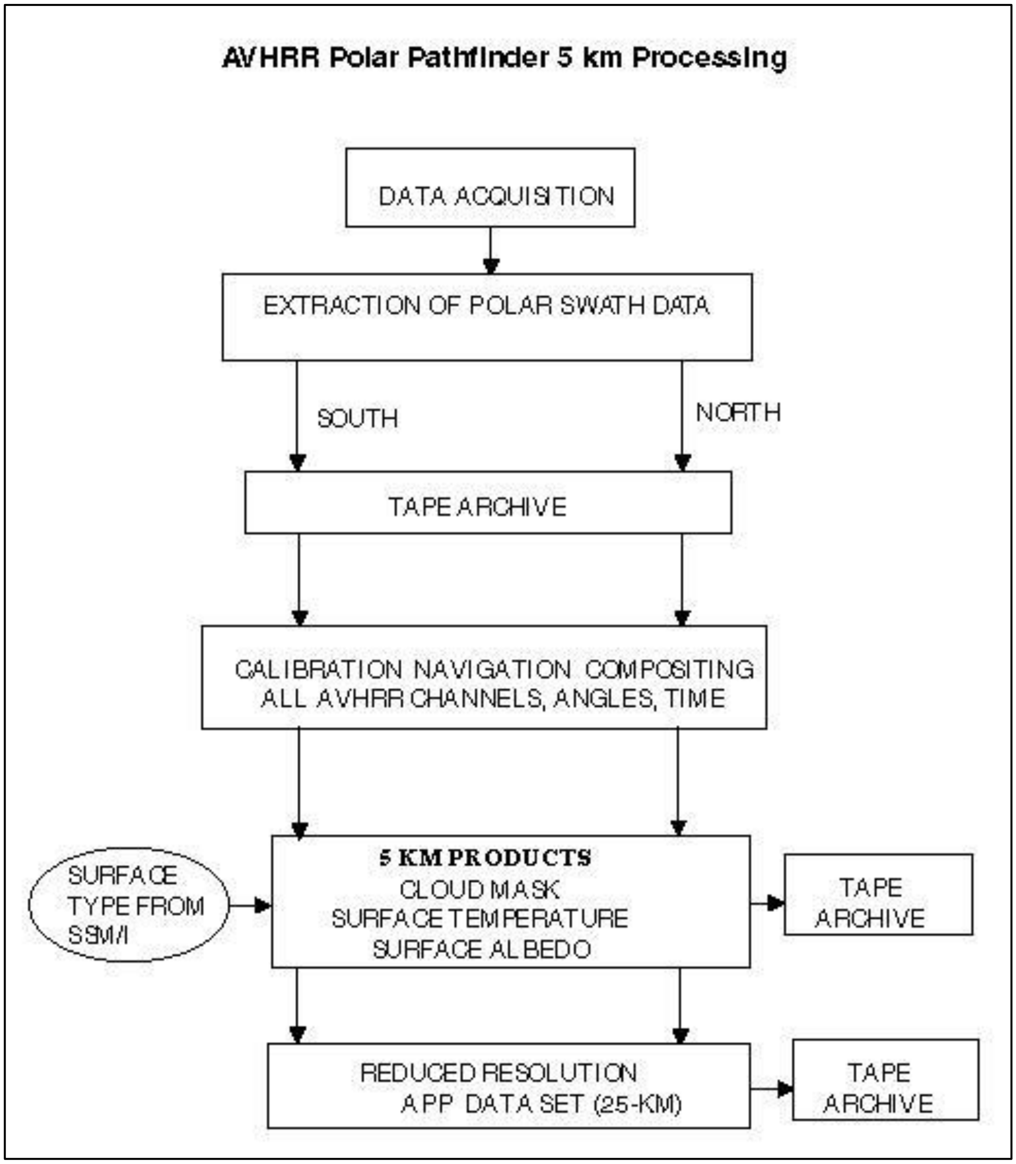

Figure 6. AVHRR Polar Pathfinder 5 km Processing

2.3 Processing

2.3.1 Processing Steps

Figure 6. AVHRR Polar Pathfinder 5 km Processing summarizes the generation of 5 km AVHRR Polar Pathfinder products.

1. **Extraction of Polar Subsets**: Processing begins by extracting the high-latitude subsets from the global Level 1B GAC orbital data. These subsets are then written in Level 1B format to 8 mm tape.
2. **Calibration and Navigation**: The level 1B GAC data were calibrated and navigated (geolocated) to the EASE-Grid. Because the data were obtained from several sources and different calibrations were performed, the calibrations were recalculated to conform to the most current methods. Satellite ephemeris data, rather than level 1b embedded coordinates, are used to determine true latitude and longitude, to ensure that common methods are used throughout the full data set. Calibration yields raw TOA reflectance values for Channels 1 and 2; and TOA brightness temperatures for Channels 3-5. Calibrated data are then navigated to earth coordinates using an orbital ephemeris model with orbit corrections (Rosborough et al. 1994). Satellite ephemeris data are obtained from the Naval Space Surveillance Center (NAVSPASUR). An inverse navigation approach is used to assign pixels to individual grid cell locations. With this approach, the geographic latitude and longitude locations of the grid are mapped to the corresponding scan line and sample in the input data. No interpolation or averaging in time or space is applied to the resulting gridded data. Geolocation accuracy is expected to be sub-pixel approximately 2 km on average for the 5 km processing. As part of the navigation step, a set of viewing and illumination angles is provided for each pixel. These angles can be used to investigate angular effects, and to test or apply alternative bidirectional reflectance distribution function adjustments for derived products. All angles are stored in a 16-bit integer as degrees times ten. Thus, each angle is known to a precision of ± 0.05 degrees for each pixel. Solar zenith angle is the angle of the sun from the vertical. Relative azimuth angle is the absolute value of the difference of the solar azimuth angle and the satellite azimuth angle. Satellite elevation angle is the angle of the satellite above the horizon. See Figure 4 for a graphical summary of these angles. As a part of the navigation step, an acquisition time is provided for each pixel. Time is stored in an 8-bit (1-byte) integer grid as hour (UTC), scaled by ten. Thus, the acquisition time is known to a precision of ± 0.05 hours (± 3 minutes) for each pixel.
3. **Composites**: There are typically 14 satellite passes each day. At the higher latitudes, there is multiple coverage at each ground location. Both poles are within the data swath during each orbit. Orbits are combined into twice-daily composites. Data are extracted from navigated imagery to match a set of decision criteria based on the satellite scan angle and the acquisition time in relation to target times. These criteria are selected to choose data that minimize scan angles to reduce algorithm errors due to atmospheric path length and bidirectional reflectance. The selected data for the five AVHRR channels are then assembled into two composites per day, for local times of 0200 and 1400 for the Southern Hemisphere, and 0400 and 1400 for the Northern Hemisphere. The precise local time for 90 percent of the cells in each grid is within one hour of the target time, and 90 percent of the cells are acquired with scan angles less than 25 degrees. NOTE: The orbital data are

not binned and averaged as with some polar satellite data such as SSM/I. This would have made algorithms for cloud masking, albedo, and surface temperatures much more difficult if not impossible to implement.

4. **Surface Type Mask**: Figure 8 summarizes the steps in generating the surface type mask. The surface type mask is an independent process, as described in the Derivation Techniques and Algorithms section, and is created using gridded SSM/I brightness temperature data and snow cover information. This mask is necessary for both the cloud masking and the albedo and surface temperature products.
5. **Albedo and Surface Temperature**: Described in the Derivation Techniques and Algorithms section
6. **Cloud Masking**: Derivation Techniques and Algorithms section.
7. **Time Tag**: An acquisition time is provided for each pixel. The time is stored in an 8-bit integer grid as hour (UTC), scaled by ten. Thus, the acquisition time is known to a precision of ± 0.05 hours (± three minutes) for each pixel.
8. **Latitude/Longitude Values**: Grids of latitude and longitude coordinates are provided and represent the estimated center location of each 5 km pixel. Each value is stored in a 16-bit (2-byte) integer grid as degrees, scaled by 100. The resulting image is organized with the same dimensions as the Northern and Southern Hemisphere grids. For each user order, a separate set of grids is provided for latitude and longitude.

2.3.2 Processing Changes

The AVHRR Polar Pathfinder Twice-daily 5 km EASE-Grid Composites consist of several versions. Refer to Table 13.

Table 13. Version Processing Changes

Version 0	No data of this version is available. This version is only available internally and was processed using the older calibration methods.
Version 1	Calibration corrections were applied to Version 0 data Channels 4 and 5 to within 0.1 kelvin.
Version 2	Data were processed using the most currently accepted calibration techniques.
Version 3	Fixes NOAA-14 visible channel calibration problem. All future processing, including the 2001 - 2005 update, will be at this level.

2.3.3 Derivation Techniques and Algorithms

Clear sky surface skin temperature, albedo, and cloud masking are all derived from the Cloud and Surface Parameter Retrieval (CASPR) system.

1. **Channels 1 and 2**: Calibration of the shortwave visible and near-infrared channels of the AVHRR (channels 1 and 2) is originally done according to Rao and Chen (1994), which corrects for sensor drift over time. Channels 1 and 2 are normalized for mean Earth-Sun distance.
2. **Channels 3-5**: The calibration of the thermal Channels 3-5 applies non-linear functions to the satellite sensor counts to obtain brightness temperatures as described in (Walton

1998). See the Calibration section for more information. The Version 3 data uses data collected from the NOAA-16 AVHRR/3 instrument, which contains the dual Channel 3 for better cloud detection over snow- and ice-covered surfaces. For the NOAA 16 AVHRR/3 instrument, Channel 3A collects data during the day at 1.58-1.64 microns, and the values are reported in units of reflectance (0-1000). The Channel 3A reflectance is obtained by first using the Channel 4 emission temperature to estimate the thermal emission component to the total Channel 3 radiance. This thermal emission component is subsequently removed from the total radiance, leaving only the solar reflectance component (Setvak and Doswell 1991).

3. **Clear Sky Surface Skin Temperature**: Two algorithms are employed to convert brightness temperature to clear sky surface skin temperature: one for high-latitude ocean and snow-covered land, and another for snow-free land (Key 1999). For retrieval of clear sky skin temperature, a simple regression model is used to correct for atmospheric attenuation. Refer to Figure 7.

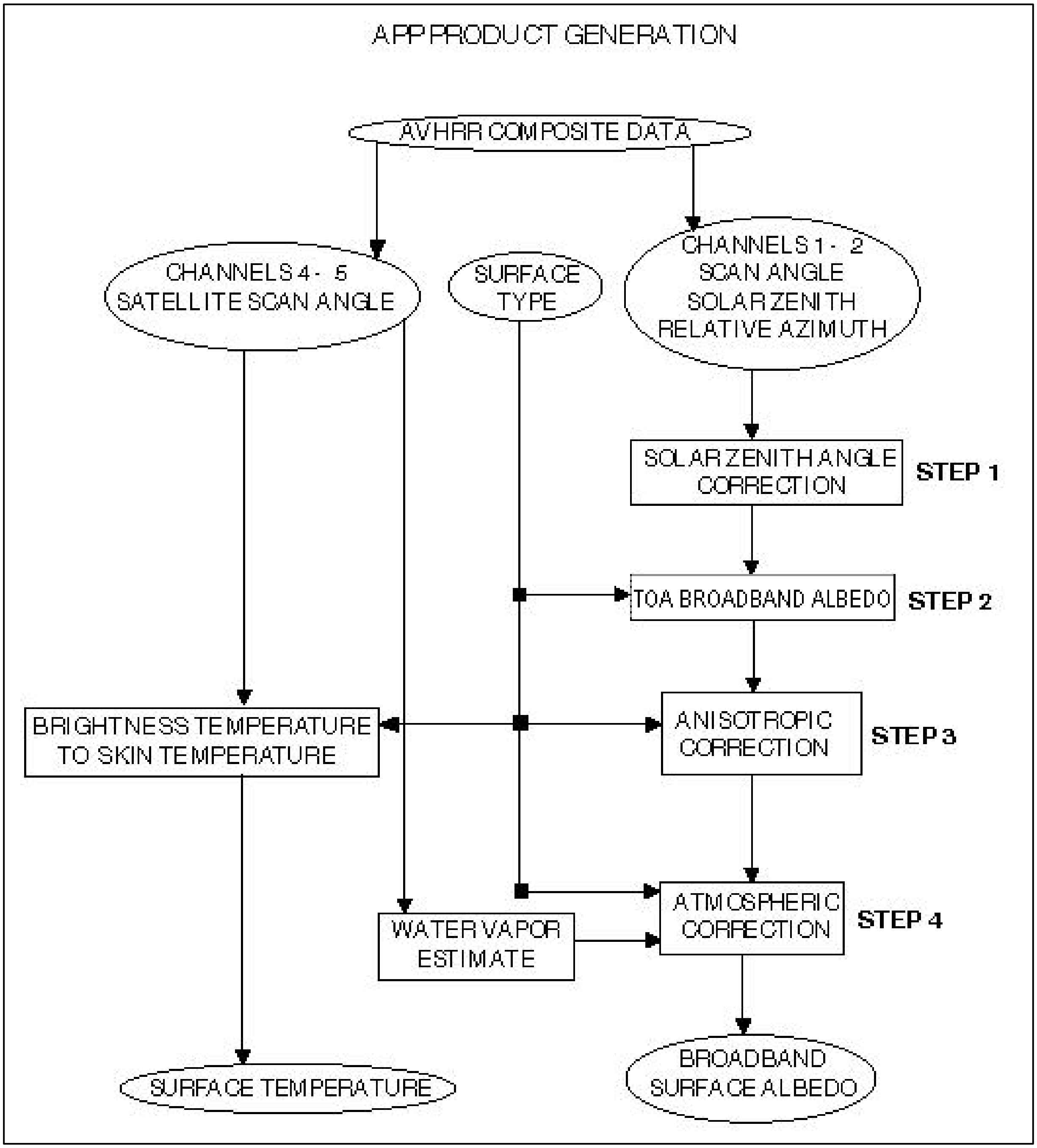

Figure 7. AVHRR Polar Pathfinder Product Generation

For high-latitude ocean and snow-covered land, the following equation applies:

$T_S = a + bT_4 + c(T_4 - T_5) + d[(T_4 - T_5)(\sec\{q\}- 1)]$

Where:

- T_S = surface temperature
- T_4 = brightness temperature in channel 4
- T_5 = brightness temperature in channel 5

- q = sensor scan angle
- a, b, c, and d = regression coefficients

To determine the empirical relationship, radiosonde data from drifting ice and land stations in the Arctic and Antarctic were used with a radiative transfer model to simulate sensor brightness temperatures. Surface temperature retrieval methods for sea ice and snow, snow-free land, and open ocean are described in detail in Key et al. 1997. The coefficients are a function of temperature range, hemisphere, and the NOAA satellite that acquired the data. Algorithms are provided in Appendix B: Atmospheric and Anisotropic Reflectance Correction Source Code Document.

Bare land is treated separately with:

$T_S = a + bT_4 + cT_5 + e*emiss4 + f*emiss5$

Where:

- emiss4 = surface emissivity in channel 4 (0.985)
- emiss5 = surface emissivity in channel 5 (0.975)

4. Clear Sky Surface Broadband Albedo

Refer to Figure 4. The retrieval of surface albedo involves four steps:

- normalize channels 1 and 2 with respect to solar zenith angle
- convert channels 1 and 2 narrowband reflectance to a broadband reflectance
- correct the TOA broadband reflectance for anisotropy
- convert the TOA broadband reflectance to a clear sky surface broadband albedo.

The general methodology in Steps 2 through 4 was used by Csiszar and Gutman (1999) for global land studies. The albedo provided here is a directional-hemispherical, apparent albedo, where apparent albedo is measured by up- and down-looking radiometers in the field.

Step 1: Normalize Channels 1 and 2 with respect to the solar zenith angle:

$P_{1,toa}= C1 / 100 * cos(zen)$

$P_{2,toa}= C2 / 100 * cos(zen)$

Where:

- C1 = percent reflectance for channel 1

- C2 = percent reflectance for channel 2
- zen = solar zenith angle
- $P_{1,toa}$ = channel 1 reflectance
- $P_{2,toa}$ = channel 2 reflectance

Step 2: Convert the narrowband reflectances in Channels 1 and 2 to a broadband reflectance. The narrow-to-broadband conversion takes the form:

$$P_{toa} = a + bp_{1,toa} + cp_{2,toa}$$

Where:

- $P_{1,toa}$ = channel 1 reflectance
- $P_{2,toa}$ = channel 2 reflectance
- P_{toa} = broadband TOA reflectance
- a, b, c = regression coefficients (Refer to Table 14)

Table 14. Regression Coefficients

Parameters	a	b	c
Bare Land	0.0404522	0.545025	0.299113
Snow/Ice	0.0215773	0.277479	0.506755
Open Ocean	0.0	1.0	0.0

To develop the regression relationship, the radiative transfer model Streamer is used to simulate the TOA reflectances over a broad range of viewing and illumination angles, atmospheric conditions, and surface types and albedos (Key and Schweiger 1998). Separate sets of coefficients are determined for different surface types. Refer to Appendix B: Atmospheric and Anisotropic Reflectance Correction Source Code Document for details.

Step 3: Correct for the dependence of the sun-satellite-surface geometry on reflectance. This is done with data presented in Suttles et al. (1988) who used the Earth Radiation Budget Experiment (ERBE) and Geostationary Operational Environmental Satellite (GOES) data to determine TOA Anisotropic Reflectance Factors (ARFs) for the broad shortwave band over various surfaces. To convert the directional reflectance to albedo, the ERBE/GOES ARFs are used:

$$a_{toa} = P_{toa} / f$$

Where:

- P_{toa} = reflectance observed at the sensor simulated by Streamer in Step 1

- f = anisotropic reflectance factor atoa = TOA albedo. The f factor is derived from a tri-linear interpolation of two tables: rmatrx and albmn.
- rmatrx: Anisotropic reflectance values based on ERBE and GOES. Dimension is (3,10,7,8) with three scene types, ten solar zenith bins, seven viewing zenith bins, and eight relative azimuth bins. See Appendix A.
- albmn: Normalizing factors for albmn. Dimension is (3,10) with three scene types and ten solar zenith bins. See Appendix A.

Preliminary results show that under certain circumstances, the reflectance of mixed pixels of open ocean and ice could have incorrect values. A Channel 1 value less than 0.0 indicates open ocean, and a value greater than 0.3 is pure ice. Values between these limits indicate pixels with mixed surface types. The final value is a weighted average of the two. Algorithms are provided in the Appendix A

Step 4: Finally, the apparent clear sky surface broadband albedo is estimated with a regression relationship of the form:

surface_albedo = $(a_{toa} - a) / b$

Where:

- a and b are a function of water vapor, aerosol amount, and solar zenith angle. The coefficients were determined with Streamer for a variety of surface and atmospheric conditions.

For open ocean, a simpler approach is taken:

surface_albedo = $a + b*a_{toa} + c*\cos(zen) + d*pw + e*aertau$

Where:

- a, b, c, d, e = coefficients determined empirically using modeled albedos
- pw = precipitable water
- zen = solar zenith angle
- aertau = atmospheric aerosols

Details are provided in Appendix A

For both cases, the aerosol optical depth is set to 0.06. Also, for both cases, the water vapor is estimated from Channels 4 and 5 using the formula:

$PW = \exp[b0 + b1(T4\text{-}T5) + b2(T5)] \cos(theta)$

Where:

- pw = precipitable water
- theta = scan angle
- b0, b1, b2 = coefficients determined over a range of surface temperatures and water vapor amounts using AVHRR radiances modeled with LOWTRAN-7. Arctic rawinsonde data were employed.
- b0 = -10.4974
- b1 = 0.751008
- b2 = 0.0453005

Details are provided in Appendix C: Retrieval of Precipitable Water Source Code Document. The calculated surface albedo is an apparent albedo, one that is measured with radiometers and which varies with changes in atmospheric conditions, particularly for bright surfaces.

5. Surface Type Mask

Refer to Figure 8 for a flowchart that summarizes the steps in generating the surface type mask.

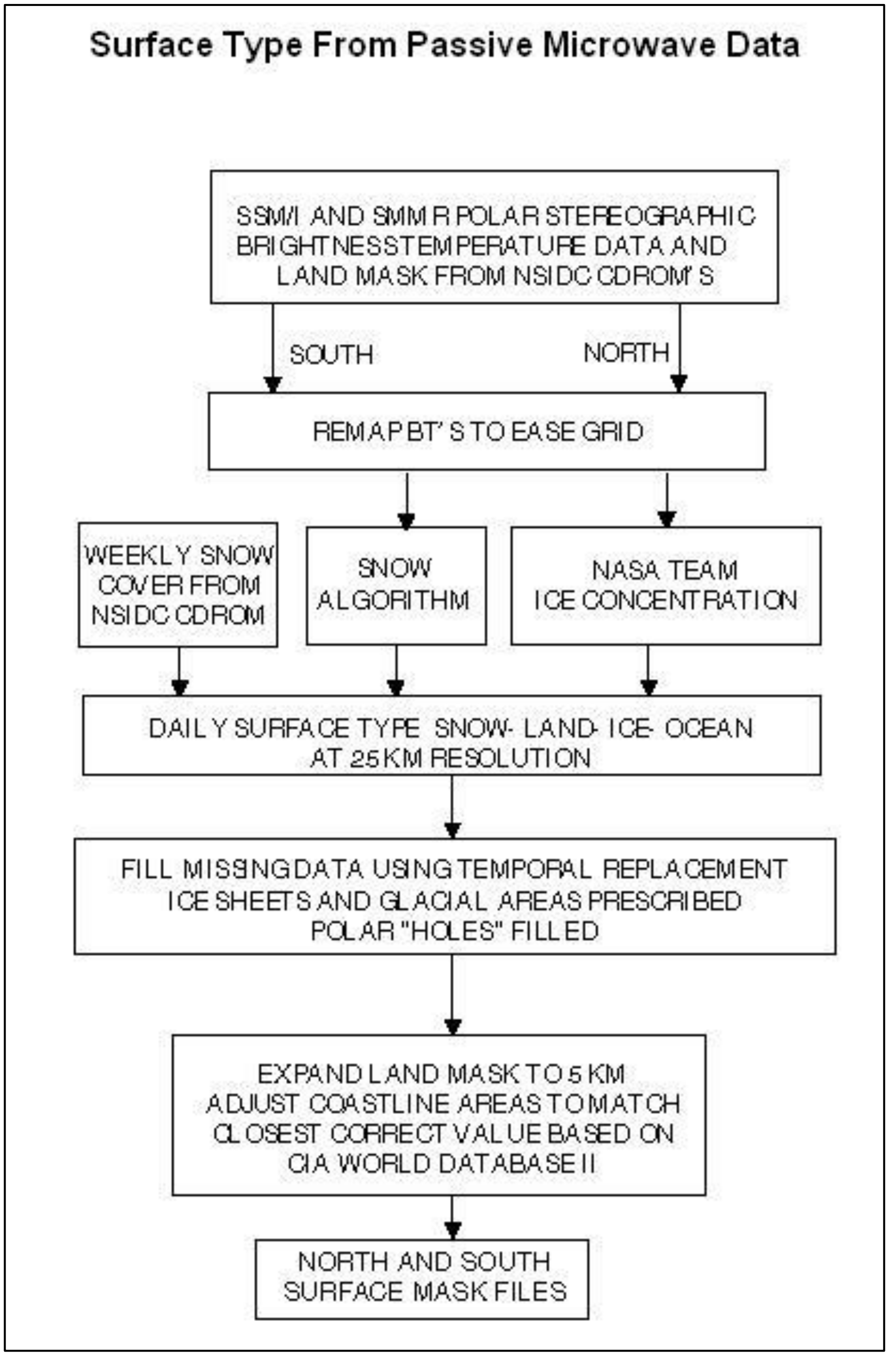
Surface Type From Passive Microwave Data
SSM/I AND SMMR POLAR STEREOGRAPHIC BRIGHTNESSTEMPERATURE DATA AND LAND MASK FROM NSIDC CDROM'S
SOUTH
NORTH
REMAP BT'S TO EASE GRID
WEEKLY SNOW COVER FROM NSIDC CDROM
SNOW ALGORITHM
NASA TEAM ICE CONCENTRATION
DAILY SURFACE TYPE SNOW- LAND- ICE- OCEAN AT 25 KM RESOLUTION
FILL MISSING DATA USING TEMPORAL REPLACEMENT ICE SHEETS AND GLACIAL AREAS PRESCRIBED POLAR "HOLES" FILLED
EXPAND LAND MASK TO 5 KM ADJUST COASTLINE AREAS TO MATCH CLOSEST CORRECT VALUE BASED ON CIA WORLD DATABASE II
NORTH AND SOUTH SURFACE MASK FILES

Figure 8. Surface Type from Passive Microwave Data

SSM/I brightness temperatures, regridded from the daily-averaged SSM/I Daily Polar Gridded Brightness Temperatures data set to the 25 km EASE-Grid, were first partitioned into land, including ice sheets or ocean. Over oceans, the NASA Team Sea Ice Algorithm was used to generate first-year and multi-year ice concentrations. For the purposes of the cloud detection procedure, all ocean pixels with an estimated ice concentration greater than zero percent were flagged as sea ice. In some cases, atmospheric effects and wind roughening of the ocean cause false ice concentration retrievals over open ocean. Such areas will appear as incorrectly mapped areas of sea ice in the mask; however, for cloud detection, preliminary work suggests that it is important to avoid excluding areas with less than 15 percent true ice cover. Areas that consist of at least 50 percent multi-year ice are assigned the multi-year flag. The multiyear ice estimate is subject to considerable error and uncertainty, particularly during periods when surface melt is likely. At such times, characteristic multiyear ice emission properties are partially or completely hidden, and melt-freeze metamorphosis or other changes in the snow cover on sea ice may cause first-year ice to appear as a fraction of multi-year ice. Since thresholds were applied to the output from ice concentration, ice type, and snow depth algorithms, slight variations in the values yield a noisy appearance in the resulting binary-flag images, particularly in spring and summer due to algorithm limitations.

Over land, snow cover is detected with an SSM/I algorithm described in Goodison (1989). A gradient value (19 GHz horizontal polarization minus 37 GHz horizontal polarization) greater than zero indicates snow, and the pixel is assigned a snow surface type. No special thresholds are applied, nor are special cases such as melting snow addressed in this mask. The SSM/I data have different coverage areas, and thus the regridding process of the SSM/I leaves some areas without information over certain regions. The Near-Real-Time SSM/I EASE-Grid Daily Global Ice Concentration and Snow Extent and Northern Hemisphere EASE-Grid Weekly Snow Cover and Sea Ice Extent products from NSIDC were used to fill in the missing regions wherever possible.

The resulting 5 km mask is over-sampled from the 25 km SSM/I data. Because coastal areas have a jagged appearance, a 5 km land mask is applied to the surface mask, and the nearest appropriate surface type mask is used.

The SSM/I-based mask is not intended to serve as a substitute for surface information from standard products such as the formal SSM/I Pathfinder snow or sea ice concentration data sets. It should only be used as an indication of the general surface conditions. Users interested in detailed snow and sea ice parameters should consult the standard products listed above. Table 15 lists the cell value for various surface types.

Table 15. Cell Values for Surface Types

Cell Value	Description
10	Open water
20-29	Predominately first-year ice
30-39	Predominately multiyear ice
40	Bare land
50	Snow-covered land
60	Ice sheet

Ranges for first-year and multiyear ice types indicate the predominant ice type and the total ice concentration in tens of percent. For example, a value of 24 means that first-year ice predominates and that the total ice concentration is between 41 and 50 percent.

6. Cloud Masks

Three cloud masks are provided since each method has advantages under certain conditions. One is based on multi-channel, multiday techniques in the CASPR algorithms; another is obtained by using a long time series of Channel 4; and a third is a modified method of replacing the Channel 4 statistics required in the CASPR algorithm with the Channel 4 time series from the second method. Cloud masks are created separately for the morning and afternoon composites. Users can choose which combinations of cloud masks to use; thus, providing some control over the confidence in the presence of clouds. This may also provide some insight into the types of clouds as well since each algorithm has somewhat different sensitivities.

a. CASPR Method

CASPR has two modes for cloud masking. The first operates on data from a single day using a combination of spectral tests and thresholding and can be adjusted for greater or less sensitivity to clouds. The second method also uses the results from the single day cloud masking in a cloud conservative mode, but comparisons are also done between consecutive days to ensure clear pixels. A time series of mean values for each channel is generated, and final cloud masking is performed by thresholding the data with the mean values. The multiday method is used with Versions 1 and 2 data (bit 0), while single-day and multiday methods are both used with Version 3 data (bits 0 and 1, respectively). The following discussion describes the details of the CASPR algorithm used to produce the Version 1 and Version 2 bit 0 cloud mask. The algorithm was somewhat modified for Version 3, and is described in CASPR User's Guide, Part II: Reference.

Cloud detection methods in CASPR are optimized for polar conditions, where cloud screening is complicated by the low contrast between clouds and surface in reflected and thermal bands, and by the lack of solar illumination at night and for part of the year. The basic detection procedure involves a combination of split-window and single channel thresholds and temporal change detection using estimates of clear sky surface temperature and albedo determined separately as a function of surface type. SSM/I data are used to partition the surface into open water, ice, snow-free land, and snow-covered land with supplementation by a high-resolution coastline and an AVHRR-based surface type within 50 km of a coast. This allows tighter optimization of the thresholds since the general surface type is known.

The cloud masking procedure consists of thresholding operations based on modeled sensor radiances. AVHRR radiances are simulated using the Streamer algorithm for a wide variety of surface and atmospheric conditions, and values that approximately divide clear from cloudy scenes are determined. The daytime and nighttime algorithms have very different logic. In the presence of sunlight (daytime), spectral cloud tests work relatively well because false detections labeling clear pixels as cloudy are infrequent. The daytime algorithm uses only spectral tests and is thus conceptually simple: initialize the cloud mask to clear, then apply the cloud tests to label cloudy pixels.

In the absence of sunlight, the procedure is different because there is less spectral information available, and because AVHRR Channel 3 tends to be noisy at low temperatures, which is common during the polar night. Under most conditions, optically thick clouds simply cannot be distinguished from clear sky with the three thermal channels alone. The nighttime algorithm first applies the spectral cloud tests, resulting in some false detections for noisy pixels and for optically thick clouds. The spectral test thresholds are set to be clear-conservative to increase the likelihood that those pixels labeled as clear are in fact clear. Next, a clear sky channel 4 temperature image is constructed by passing a mean filter over the clear pixels, eliminating much of the noise and false detections. Another threshold test (based on the clear sky map) is applied to the image, such that temperature deviations greater than the threshold from the clear sky mean signify that a pixel is cloudy. In this final test, it is possible for the cloud to be either warmer or colder than the surface, especially during the winter.

Most of the cloud test concepts are found in the Air Force SERCAA procedures (Gustafson et al. 1994). Some appear in the NOAA CLAVR algorithm (Stowe et al. 1991), while most were developed and/or used elsewhere. In the discussion below, warm cloud refers to clouds with higher temperatures than the surface, while cold clouds have temperatures lower than the surface. The cloud tests are as follows:

i. Cirrus test (11 µm - 12 µm, day and night): A pixel is cloudy if BTD45 is greater than CCT_TABLE, where BTD45 is channel four minus channel five brightness temperature difference, and CCT_TABLE is a table of thresholds from (Saunders and Kriebel 1988) extended to include lower temperatures (range: 230-310 kelvin). The thresholds are the expected maximum BTD45 for clear sky.

ii. Warm clouds test (11 µm - 12 µm, day and night): For warm clouds BTD45 may be negative, so an additional test is required (Yamanouchi et al. 1987). A pixel is considered to be cloudy if BTD45 is less than WARM45. This test is most applicable at night in the presence of a temperature inversion.

iii. Water cloud test (3.7 µm, day only) - A pixel is considered cloudy if REF3 is greater than REF3_THRESH and REF1 is greater than REF1_THRESH, where REF1 and REF3 are channels one and three reflectances, and REF1_THRESH and REF3_THRESH are threshold values.

iv. Low stratus and thin cirrus tests (3.7 µm - 11 µm, night only): A pixel is cloudy if BTD34 is less than or equal to LSTTCI_34LO, or BTD34 is greater than or equal to LSTTCI_34HI; where BTD34 is the brightness temperature difference between channels three and four.

The above tests are done in a clear-conservative mode, meaning that the threshold is set so that values are more likely to be clear. Next, the following three steps are done:

i. Examine the time change in spectral characteristics between pairs of days. A pixel is clear if the spectral tests say it is clear, and if it varies little on a daily basis. The Channel 4 brightness temperature and the adjusted brightness temperature difference (4 minus 5) are used in the time tests when any part of the image is dark. The Channel 4 brightness temperature, Channel 1 reflectance, and Channel 3 reflectance are used during the day.

ii. Clear sky statistics are compiled for each pixel using all clear values within a distance of the nine pixels and over five days. Means and standard deviations of the Channels 1 and 3 reflectances (day only) and the Channel 4 temperature (day and night) are computed. If a cell occurs along a coastline, only pixels with the same surface type (LAND or NOTLAND) as the center pixel are used. Pixels in the clear sky composite that did not have means and standard deviations assigned, due to the lack of clear values, are filled by bilinear interpolation.

iii. The clear sky composite statistics are used in a threshold operation to label or relabel all pixels as clear or cloudy. A pixel can be labeled as cloudy if its brightness temperature is higher than its mean clear value plus some number of standard deviations; or if its reflectance in Channel 1 or 3 is higher than its mean clear reflectance plus some number of standard deviations. Similarly, a pixel can be labeled as cloudy if its brightness temperature is lower than its mean clear value minus some number of standard deviations. If any part of the image is dark, the nighttime final thresholding is calculated for the entire image, though both the daytime and nighttime spectral cloud tests are used in the initial labeling. Refer to Step 1.

The only distinction between surface types is land and ocean. Land may be snow-free or snow-covered. Ocean can be open water or sea ice. Additionally, the nighttime procedure does not incorporate any surface type distinction. Comparisons with SHEBA surface observations of cloud amount (synoptic observations every three hours) yield a bias of 0.1 and a root mean square error (RMSE) of 0.3.

b. Time Series of Channel 4 Data

Note: This method is used for Versions 1 and 2 data only. The second cloud masking method makes use of long time series of Channel 4 data. Beginning with at least 120 days of composited Channel 4 brightness temperatures averaged to 25 km pixels, a combination of median and maximum filtering is applied to 20-day increments of the brightness temperatures for each pixel. The filtering selects the most likely clear-sky temperature for each pixel, and interpolates time to fill in the estimated temperature during cloudy periods. This is assumed to be close to a clear-sky time series of channel four brightness temperatures at each pixel location. A cloud mask is generated by thresholding the actual channel 4 with the time series. Separate time series are created for the morning and afternoon composites, since solar heating may change the mean temperatures.

c. Modified CASPR Cloud Masking

Note: The third cloud mask is done exactly the same as the first method using the CASPR routines, except that the Channel 4 statistical mean is replaced with the Channel 4 values from the time series algorithm described above. Because a cloud can be persistent, the longer time series may provide a better mean value. This is especially critical during the periods of low or no solar illumination when the visible spectral tests are not available for the cloud masking.

7. Time Mask

An acquisition time is provided for each pixel. The time is stored in an 8-bit (1-byte) integer grid as hour (UTC), scaled by ten. Thus, the acquisition time is known to a precision of ± 0.05 hours (± three minutes) for each pixel.

8. Reduced-Resolution Products

To assist users in selecting 5 km data, or for applications that require a lower data volume product, reduced-resolution data are available for the five AVHRR channels, clear sky surface broadband albedo and skin temperature, average albedo and temperature, valid fraction file, solar zenith angle, satellite elevation angle, sun-satellite relative azimuth angle, surface type mask, cloud mask, cloud fraction files, and Universal Coordinated Time (UTC) of acquisition. These products are subsampled to a grid resolution of 25 km by 25 km. Please see the AVHRR Polar Pathfinder Twice-Daily 25 km EASE-Grid Composites documentation for details and ordering.

9. Discussion of Clear Sky Surface Broadband Albedo and Clear Sky Surface Skin Temperature Algorithms

These algorithms are appropriate for clear-sky conditions only. Nonetheless, the algorithms have been applied to all pixels, not just to pixels flagged as clear-sky. The reasoning for this is three-fold: First, the cloud detection algorithm continues to evolve and improve, so different versions of the cloud mask can be provided to users without negating the value of the surface products for clear-sky pixels. Second, users are encouraged to apply their own cloud detection procedures (including manual interpretation), which can be tuned to maximize performance for specific regions and case studies. These custom cloud masks can then be applied to the AVHRR Polar Pathfinder 5 km clear sky surface skin temperature and surface albedo product, eliminating the need for users to implement their own surface product algorithms. Third, some users may wish to investigate the usefulness of the surface products under marginal cloud conditions that may be detected by AVHRR as being cloudy, but which may actually consist of very thin cloud or diamond dust precipitation. Since the CASPR surface algorithms do not work properly for cloudy pixels, the AVHRR Polar Pathfinder 5 km clear sky skin temperature and albedo grids contain inaccurate values. As noted above, users need to apply the cloud mask provided in this data set or use their own cloud detection method.

Evaluation of the CASPR surface products and the cloud detection approach is ongoing, and has identified several shortcomings of the existing products. The CASPR algorithms are optimized for performance over sea ice. Tests indicate that performance of the standard algorithm over ice sheets warrants improvement. The accuracy of the model simulations that form the basis of the CASPR albedo algorithm decreases with low solar zenith angles, such as the sun being low on the horizon. Thus, albedos calculated in spring and autumn are likely to be less accurate than those for periods with higher sun angles. Also, parameterizations used to address bidirectional reflectance are not equally representative of all snow conditions for a full range of sun and satellite viewing angles.

The performance of the clear sky surface skin temperature algorithms was evaluated using surface observations from an ice camp in the Beaufort Sea north of Alaska during March and April, 1992, and from an annual cycle of data from a land site near Barrow, Alaska (Key et al. 1997). The bias and RMSE at the ice camp were -0.30 degrees and 0.61 degrees, respectively. At the land site, the bias and RMSE were -0.97 degrees and 4.70 degrees, respectively.

The surface type used to calculate clear sky surface broadband albedo is the same as that used to calculate clear sky surface skin temperature. Values for albedo are neglected where the solar zenith angle is greater than or equal to 85 degrees. An annual cycle of data from Barrow, Alaska was used to validate the clear sky surface broadband albedo (Key et al. 1997). The bias and RMSE are 0 percent and 17 percent, respectively, at the land site. However, errors are expected to be smaller over ocean, sea ice, and snow.

2.4 Quality, Errors, and Limitations

A known problem exists with the Version 1 and 2 products. When the start of Acquisition Time (UTC) for an orbital swath started near the end of a UTC day, for example, 1983, day 122, 23:54, the end of the swath would extend into the beginning of the following day, for example, 1983, day 123, 00:06. As a result, the corresponding next day pixels are included in the current day composite. This is not a problem in and of itself. However, the UTC parameters for these next day pixels contain hour values of zero and greater, for example, 1983, day 123, 0.1. This causes the pixels to appear to have an acquisition time 24 hours earlier than their true acquisition time. Version 3 data labels next day pixels with a time of acquisition value of 24 or greater, for example, 1983, day 123, 24.1 from the preceding example.

A known problem exists with the NOAA-16 Version 3 data. Due to a problem with the scan motor, all the channel data are shifted sporadically causing the channels to contain data from another channel; thus, the derived parameters are in error when this channel shift occurs. The dates when the scan motor problem affects the data are list in Table 16Table 16. Dates of Bad Data. Note however that the scan motor problem does not affect the entire composited image on these dates, but rather there are patches of bad data within the composite. See Figure 9 for an example. For more information regarding the scan motor problem, please visit the NOAA Satellite and Information Services: Office of Satellite Operations NOAA 16 AVHRR Subsystem Summary web page.

Table 16. Dates of Bad Data

Year	Dates
2001	1 January (missing files)
	6, 23, 31 January 1, 13, 19, 27 February 13, 14 March 3, 8, 9, 28 April 23 June 22 July
2002	16 August 31 October 11 December
2003	11, 17, 22 March 1 April 12 May 24 June 19 - 22, 24 September 5, 22, 26 December
2004	14 - 24 January 10, 16 - 18, 20 - 31 March 1 - 9, 12 - 14, 20 - 29 April 2 - 5, 8 - 15, 19 - 21, 29, 30 May 9 June 6, 10 - 17, 21, 23, 25, 27, 28 - 31 July 1 - 3 August 9, 20 September
2005	30 March (missing files) 21 May

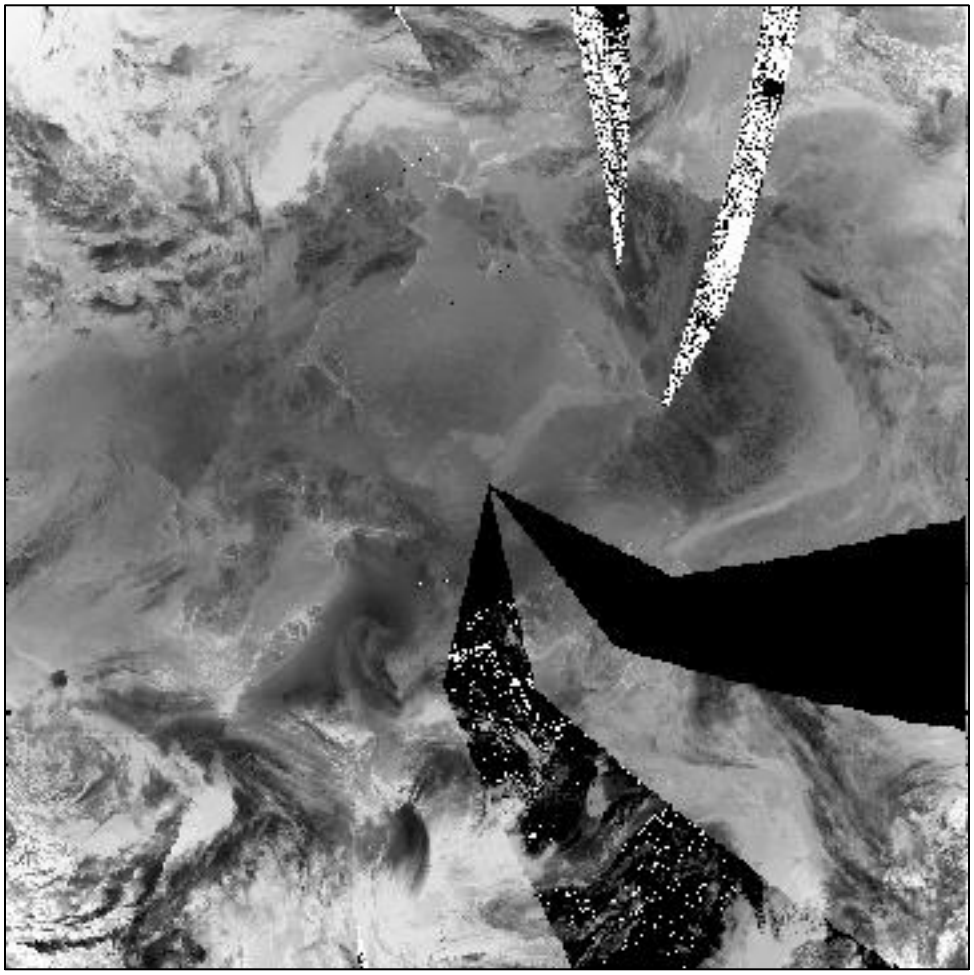

Figure 9. Northern Hemisphere Composite at 1400 Hours Showing Patches of Bad Data

2.4.1 Sensor Calibration Errors

Graphs spanning the entire temporal coverage of the data indicate that sensor calibration errors have occurred. Albedo measurements in areas where only small variances are expected, consistently show a substantial dip in 1995 and again in 2001 to 2005. This unusual pattern is most likely resulting from incorrect sensor calibration during those periods. Figure 10 through Figure 12 show the albedo dip patterns at Summit, South Dome, and Humboldt glaciers in Greenland.

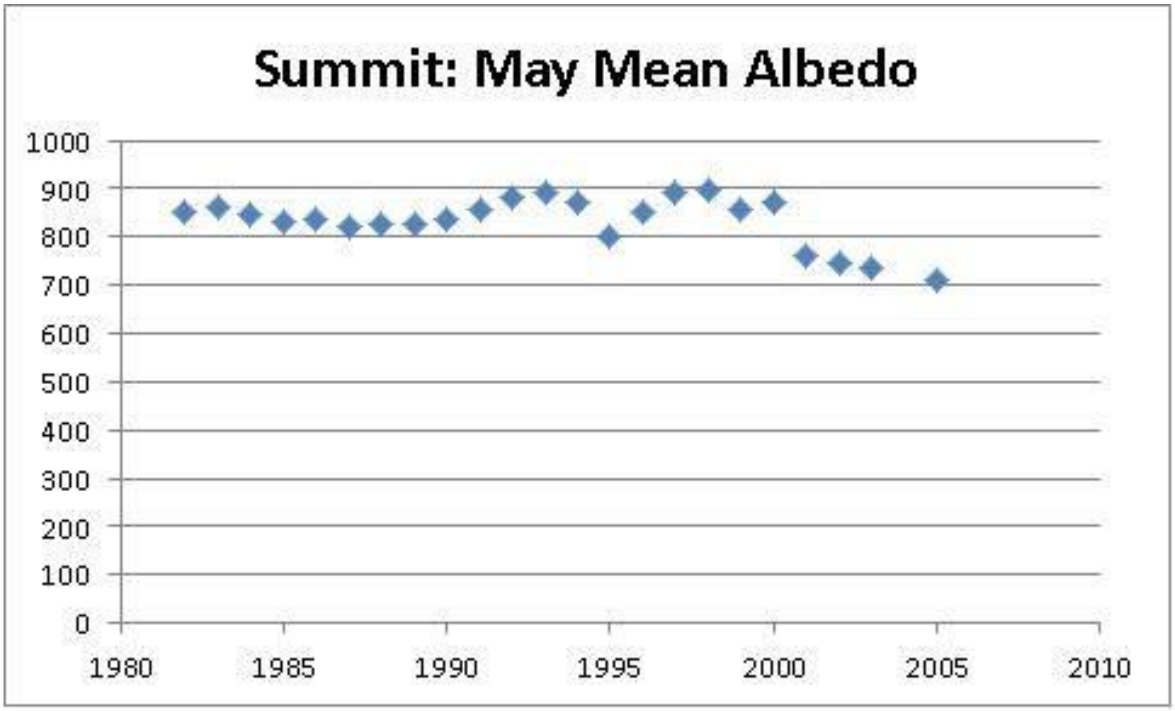

Figure 10. Albedo Measurements at Summit Glacier

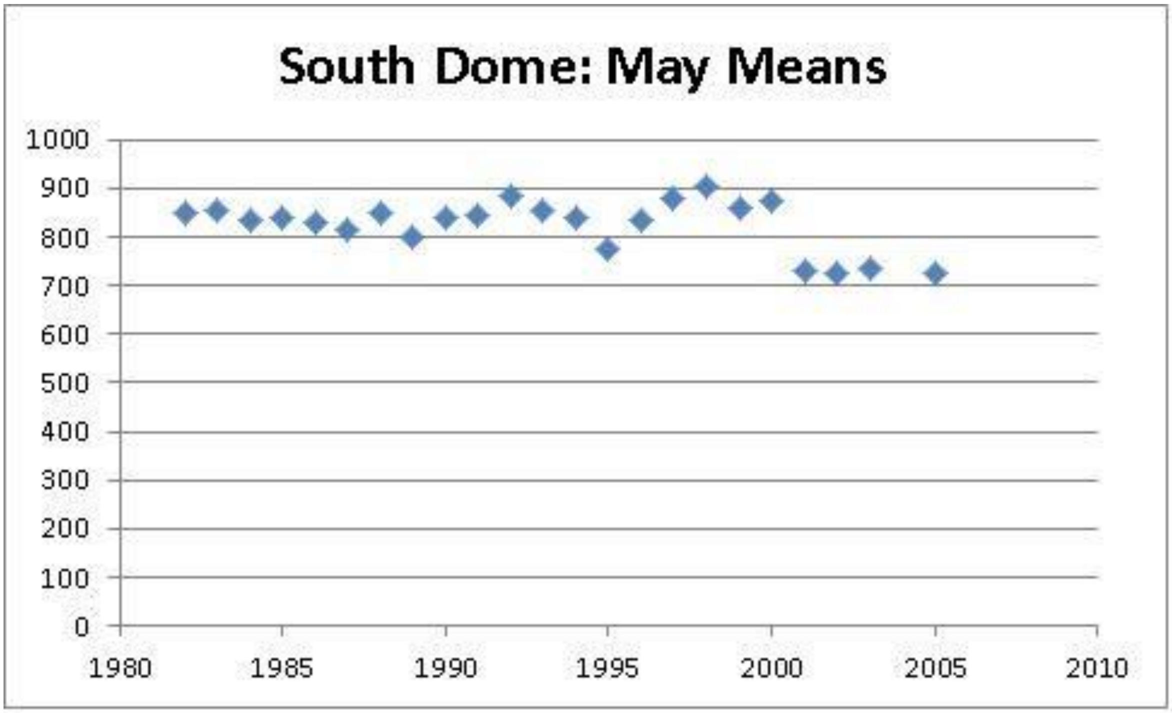

Figure 11. Albedo Measurements at South Dome Glacier

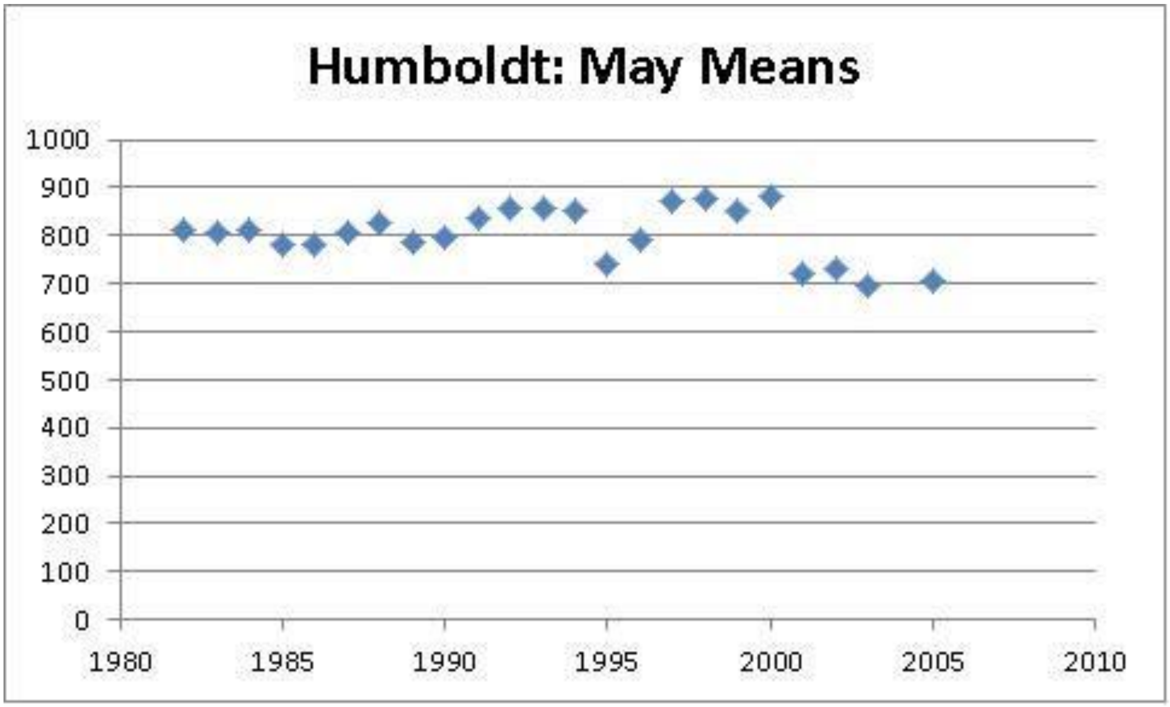

Figure 12. Albedo Measurements at Humboldt Glacier

2.4.2 Geolocation Errors

Projection inconsistencies were found between the CASPR processing Versions 2 and 3, resulting in slight geolocation variance. Instances in which the two processing versions are intermixed can cause animated images to appear to jitter. From 2002 to 2005, an incorrect ephemeris was used causing a geographic shift in the data.

2.4.3 Swath Composite Errors

The swath compositing seems to be in error between 26 July – 2 August, 1983. Starting on 15 May 2004, the swath composite shows similar errors, which increase in intensity throughout the 2004 summer months.

2.4.4 Data Identifier Errors

Isolated instances of misidentified data parameters were found. In these cases, either the land masking or the IR/Vis channels is interchanged.

2.4.5 Measurement Error Parameters

Channel data are accurate to within approximately ± 0.2 percent based on sensor noise level of 0.4 Data Number (DN). Temperatures are accurate to within approximately two kelvins. Relative albedos in adjacent grid cells are accurate to within approximately five percent. However, large errors are expected in the absolute albedo values.

2.4.6 Data Validation by Source

Product validation is a continuing process that takes advantage of comparative data as they become available. Comparisons were made between AVHRR Polar Pathfinder clear sky skin temperatures and surface-based measurements obtained at the South Pole over a seven-day period in 1995. These field data were collected by Robert Stone of the Cooperative Institute for Research in Environmental Sciences (CIRES) using a sled-mounted KT-19 pyrometer. Excluding observations when cloud cover was present, the agreement was generally within 0.5 kelvin. For data averaged over a four-hour period, temperatures were within 0.1 kelvin. A mean of -38.15 degrees Celsius for the AVHRR Polar Pathfinder observations, versus a mean of -38.25 degrees Celsius for the field data.

Evaluations were also performed for the AVHRR Polar Pathfinder retrievals of surface albedo over the Greenland Ice Sheet through comparisons with albedo measured at 14 Automatic Weather Stations (AWS) around the Greenland Ice Sheet from January 1997 to August 1998. Results show that AVHRR-derived surface albedo values are, on average, 10 percent less than those measured by the AWS stations. However, station measurements tend to be positively biased by about four percent, and the differences in absolute albedo may be less, about six percent. In regions of Greenland where the albedo variability is small, such as the dry snow facies, the AVHRR albedo uncertainty exceeds the natural variability. Stroeve concluded that while further work is needed to improve the absolute accuracy of the AVHRR-derived surface albedo, the data provide temporally and spatially consistent estimates of the Greenland Ice Sheet albedo (Stroeve et al. 2001) and (Stroeve 2002).

Analyses of the AVHRR Polar Pathfinder data, compared with data from the Surface Heat Budget of the Arctic Ocean (SHEBA) project, are in progress. See (Maslanik et al. 2000) for preliminary results. The cloud masking process was assessed and refined throughout the duration of the project to optimize the algorithm for the entire areas of coverage. Comparisons of areally-averaged cloud fractions from the AVHRR Polar Pathfinder Twice-daily 5 km EASE-Grid Composites with field observations at the SHEBA field site show that the AVHRR data were within nine percent of the cloud lidar/radar observations averaged from April to July 1998 with Pathfinder data underestimating cloud fraction relative to the field measurements. Differences in monthly means for this period ranged from 2 percent in June to 21 percent in July. Comparison of all-sky skin temperature and albedo values derived from the AVHRR Polar Pathfinder Twice-daily 5 km EASE-Grid Composites with SHEBA observations is described in (Maslanik et al. 2000).

Other validation studies of surface temperature and albedo retrieval procedures included surface observations from a NOAA research site near Barrow, Alaska, 71.32 degrees north latitude, 156.61 degrees west longitude. Daily AVHRR data from a preliminary Pathfinder data set from mid-1992 to mid-1993 were used for this validation (Meier et al. 1997). Surface temperature estimates agreed with observations, with a correlation coefficient of 0.98, a bias of -0.97, and a RMSE of 4.70. For surface albedo, the bias (mean error) in the estimates was near zero, r=0.81, bias=0.00, RMSE=0.17, but the individual observations exhibited significant variability, attributed to surface inhomogeneity and retrieval scheme sensitivity to changes in atmospheric aerosol and water vapor amounts.

Accuracies of the products are difficult to determine given the limited nature of existing case studies. Also, conditions vary substantially across the large product domains and over time. Plans are being developed to further define product accuracies for snow-covered areas, sea ice, and ice sheets. Based on studies to date, accuracies in general are approximately ± 2 kelvin for AVHRR-derived clear sky skin temperatures and ± 0.06 kelvin for albedo. Much of this error is likely due to uncertainties in the performance of the cloud detection methods. For clear sky conditions, accuracies for albedo and temperature products are expected to be in the range noted in the Greenland Barrow case studies.

2.4.7 Additional Quality Assessments

More information on the characteristics and quality of the AVHRR Polar Pathfinder data is provided by (Maslanik et al. 1998), (Maslanik et al. 2000), (Stroeve et al. 2001), and (Stroeve 2002).

2.5 Instrumentation

2.5.1 Source/Platform

The current series of NOAA Polar Orbiting Environmental Satellites (POES) has been operational since mid-1978. NOAA-7, -9, -11, -14, and -16 satellite data were used for the AVHRR Polar Pathfinder Twice-Daily 5 km EASE-Grid Composites. Refer to Table 17 for the dates of operation.

Table 17. NOAA POES Operation Dates

NOAA-7	23 July 1981 - 31 December 1984
NOAA-9	1 January 1985 - 7 November 1988
NOAA-11	8 November 1988 - 31 December 1994
NOAA-14	1 January 1995 - 31 December 2000

NOAA-16	1 January 2001 - 30 June 2005

The primary instruments aboard this third generation of satellites (TIROS-N and NOAA-6 through NOAA-16) are the AVHRR sensor and the TIROS Operational Vertical Sounder (TOVS). However, the AVHRR instrument was updated to Version AVHRR/3, which now has 6 channels. AVHRR/3 was first carried on NOAA-15 and launched in May 1998.

The ascending portion of the orbit crosses the equator at local time in the afternoon. Refer to Figure 13. The satellites are placed in orbit so that the equator crossing time is about 1400 local solar time. However, as the satellite remains in orbit, the equator crossing time shifts to later in the afternoon.

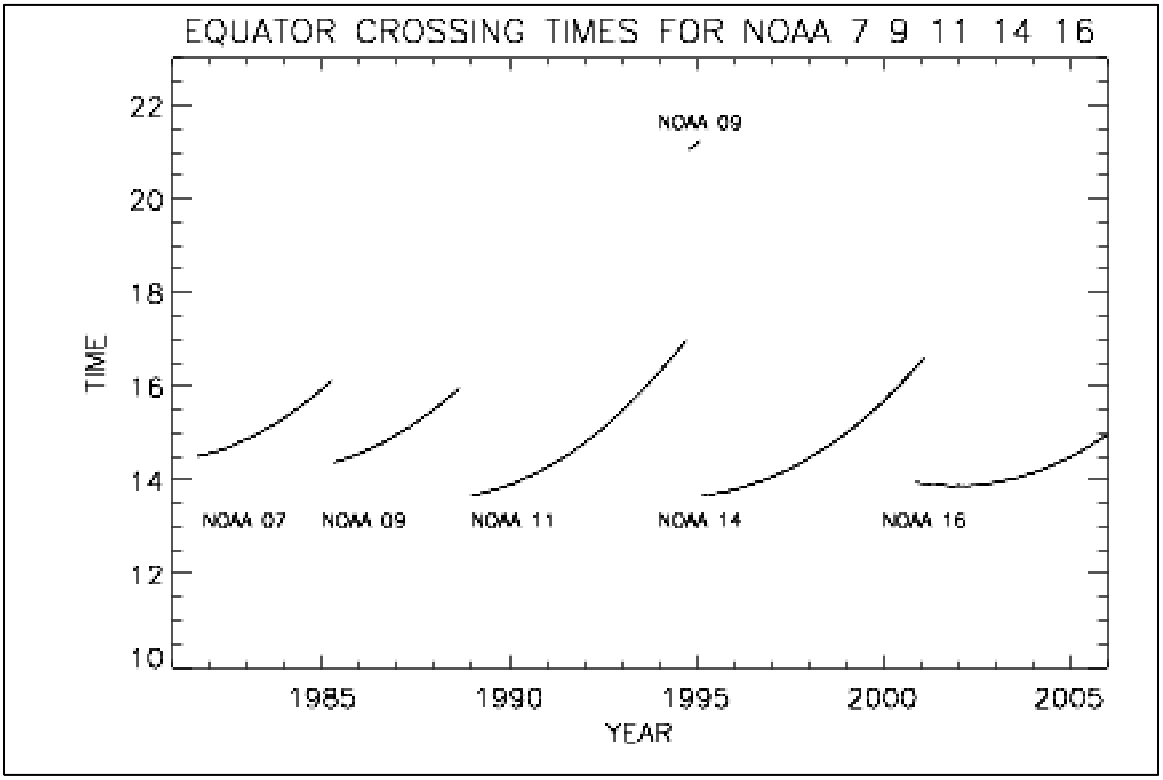

Figure 13. AVHRR Polar Pathfinder Equator Crossing Times

2.5.2 Source/Platform Mission Objectives

The AVHRR sensor was originally designed for use as an imaging radiometer for meteorological purposes, rather than for quantitative radiometric sensing (Cracknell 1997). However, as new applications evolved, quantitative radiometric data became necessary. Channels 1 and 2 were designed to provide direct quasi-linear conversion between the 10-bit digital numbers and reflectance. The thermal channels were designed to provide this conversion between the digital numbers and the temperature in degrees Celsius or kelvin. The primary reason for the introduction of the five-channel system was the need for atmospheric correction calculations in determining sea surface temperature (Cracknell 1997).

The AVHR/3 sensor on the NOAA-16 platform was designed for improved cloud detection over snow- and ice-covered surfaces.

2.5.3 Principles of Operation

The scan mirror collects earth observation data during a discrete part of the scan cycle. The scan mirror observes the scene below the spacecraft in a continuous line from horizon to horizon as it rotates. Energy from the scene is collected by a telescope and separated according to wavelength by beam splitters. Signals are amplified, filtered, and applied to the 10-bit analog/digital converter, which samples all five channels simultaneously. In Version 3, the NOAA-16 platform contains the AVHRR/3 instrument which has a dual Channel 3 for better cloud detection. Channel 3A collects data during the day at 1.6 microns, and Channel 3B collects data during the night at 3.7 microns. If the calibrated data has a value range below 1200 kelvin with a scaling factor of 10, the data is from Channel 3A. If the calibrated data has a value range of 2000 kelvin and above with a scaling factor of 10, the data is from Channel 3B. However, there are periods of time when Channel 3A is turned off. Thus, for information regarding the NOAA 16 AVHRR subsystem history, specifically the new Channels 3A and 3B, refer to the NOAA Satellite and Information Service: Office of Satellite Operations: NOAA 16 AVHRR Subsystem Summary web page.

AVHRR and AVHRR/3 Channel Wavelengths are summarized in Table 18.

Table 18. AVHRR and AVHRR/3 Wavelengths

Channel 1	0.58 µm to 0.68 µm (visible)
Channel 2	0.725 µm to 1.05 µm (reflected infrared)
Channel 3	3.55 µm to 3.92 µm (reflected/thermal infared)
Channel 3A	1.58 -1.64 µm (reflected/thermal infared, collects data during the day)
Channel 3B	3.55 -3.93 µm (reflected/thermal infared, collects data during the night)
Channel 4	10.3 µm to 11.3 µm (thermal infrared)
Channel 5	11.5 µm to 12.5 µm (thermal infrared)

The 10-bit resolution digital data is processed to create a direct readout of High Resolution Picture Transmission (HRPT) data, Automatic Picture Transmission (APT) data, 4 km GAC data, and 1 km LAC data, to ground stations throughout the world (Cracknell 1997).

2.5.4 Sensor/Instrument Measurement Geometry

The AVHRR instrument scans in the across-track direction with a continuously rotating scan mirror, viewing a swath of over 100 degrees and up to 55 degrees off-nadir. Spatial resolution is approximately 1.1 km when the view is at nadir. Scanning to 55 degrees (68 degrees satellite zenith angle relative to the earth's surface) off nadir results in a ground resolution of over 2.4 km by 6.5 km at the maximum off-nadir position (Cracknell 1997).

The AVHRR/3 is an imaging system in which a small field of view (1.3 milliradians by 1.3 milliradians) is scanned across the earth from one horizon to the other by continuous 360-degree rotation of a flat scanning mirror. The orientation of the scan lines are perpendicular to the spacecraft orbit track and the speed of rotation of the scan mirror is selected so that adjacent scan lines are contiguous at the subsatellite (nadir) position. Complete strip maps of the earth from pole to pole are thus obtained as the spacecraft travels in orbit at an altitude of approximately 833 km (450 miles). The analog data output from the sensors is digitized on board the satellite at a rate of 39,936 samples per second per channel. Each sample step corresponds to an angle of scanner rotation of 0.95 milliradians. At this sampling rate, there are 1.362 samples per Instantaneous Field of View (IFOV). A total of 2048 samples will be obtained per channel per Earth scan, which will span an angle of ±55.4 degrees from the nadir (subpoint view). All six spectral channels of the AVHRR/3 are registered so that they all measure energy from the same spot on the earth at the same time. All six channels are also calibrated so that the signal amplitude in each channel is a measure of the scene radiance. Although the AVHRR/3 has six channels, only five are transmitted to the ground at any one time. The radiometers are designed to operate within specification for a period of three years in orbit (Goodrum et al. 2000).

2.5.5 Manufacturer of Sensor/Instrument

The first AVHRR sensor was designed and built by ITT Aerospace in 1976. Subsequent instruments were built by ITT Aerospace under contract with NASA, which procured the instruments on behalf of NOAA (Cracknell 1997).

AVHRR/3 was designed and built by ITT Industries' Aerospace/Communications Division (A/CD), and it was first carried on NOAA-15 and launched in May 1998.

2.5.6 Calibration

When the AVHRR Polar Pathfinder program began, calibrations were based on the NOAA Polar Orbiter Data (POD) User's Guide (Kidwell 1995). Since then, at least four different publications have presented different methods of calibrating AVHRR data. Even with extensive pre-launch testing and calibrations, satellite sensors change over time, and improved methods are developed.

In some cases, particularly for the visible channels, the sensor characteristics are not fully predictable, and the satellite must be in orbit before changes can be detected to make the proper corrections. However, the calibrations for this data set follows the guide lines in two publications: NOAA Polar Orbiter Data User's Guide: Section 3.3 Calibration of AVHRR Data was used for NOAA 7 through NOAA 14 satellite data, and NOAA KLM User's Guide: Section 7.1 AVHRR was used for the NOAA 16 satellite data.

2.5.7 Visible and Near Infrared Channels

The POD calibration method for Channels 1 (visible) and 2 (near-infrared) used pre-launch values. The primary use of these channels has traditionally been for vegetation studies over land. Investigations revealed that the two channels drifted from the initial launch conditions. A set of time-varying coefficients was subsequently developed to correct for the drifting sensors. In 1999, the calibration for NOAA-14 changed and new time-varying equations were recommended (Rao and Chen 1999). See the Derivation Techniques and Algorithms section of this document for more information.

2.5.8 Thermal Channels

At the start of the AVHRR Polar Pathfinder program, calibration of the thermal channels followed the guidelines in the POD. The methodology is as follows:

- compute a linear slope and intercept based on the instrument count of space and internal blackbody
- apply the slope and intercept to calculate scene radiance
- convert the radiance to brightness temperature using the Plank functions
- apply a non-linear correction to the temperatures using a look-up table of corrections based on scene and blackbody temperature

This method was optimized for a narrow range of AVHRR sea surface temperature products. The Walton method describes an effective method for correcting AVHRR thermal channels 3, 4, and 5 to accommodate the wide range of temperatures in the polar regions. The NOAA/NASA Land Pathfinder group also selected this method, and it is used for the AVHRR Polar Pathfinder 5km Data Set.

The Walton method applies a non-linear correction to the radiance, which is then converted to brightness temperature using a quadratic function with coefficients derived from pre-flight calibrations. The result, while not optimized for the narrower range needed for sea surface temperature measurements, appears to be an improvement for polar regions.

The Walton calibration method was implemented in March 1999 for thermal channels from all NOAA AVHRR satellites. Correction tables were generated for previously processed data to closely match the new calibration techniques. These corrections appear to be compatible to within 0.1 degrees (Walton et al. 1998).

3 RELATED DATA SETS

- Extended AVHRR Polar Pathfinder (APP-x)
- Polar Pathfinder Daily 25 km EASE-Grid Sea Ice Motion Vectors

4 CONTACTS AND ACKNOWLEDGMENTS

Chuck Fowler, James Maslanik, and William Emery
Colorado Center for Astrodynamics Research
CCAR, 431 UCB
University of Colorado
Boulder, CO 80309-0431

Terry Haran and Ted Scambos
National Snow and Ice Data Center
CIRES, 449 UCB
University of Colorado
Boulder, CO 80309-0449

Jeffrey Key
NOAA/NESDIS
1225 W. Dayton St.
Madison, WI 53706

NSIDC User Services
National Snow and Ice Data Center
CIRES, 449 UCB
University of Colorado
Boulder, CO 80309-0449 USA
phone: +1 303.492.6199
fax: +1 303.492.2468
form: Contact NSIDC User Services
e-mail: nsidc@nsidc.org

5 REFERENCES

Cracknell, A. P. 1997. *The Advanced Very High-Resolution Radiometer*. London: Taylor & Francis.

Csiszar, I. and G. Gutman. 1999. Mapping Global Land Surface Albedo from NOAA AVHRR. *Journal of Geophysical Research* 104(D6):6215-6228.

Diner, D. J., J. V. Martonchik, C. Borel, S. A. W. Gerstl, H. R. Gordon, Y. Knyazikhin, R. Myneni, B. Pinty, M. M. Verstraete. 1999. *MISR Level 2 Surface Retrieval Algorithm Theoretical Basis*. Jet Propulsion Laboratory.

Goodison, B. E. 1989. Determination of Areal Snow Water Equivalent on the Canadian Prairies Using Passive Microwave Satellite Data. *IGARSS '89 Proceedings* 3:1243-1246.

Goodrum, Geoffrey, Katherine B. Kidwell, Wayne Winston. 2000. *NOAA KLM User's Guide with NOAA-N, -N' Supplement*. National Oceanic and Atmospheric Administration.

Gustafson, G. B. et al. 1994. Support of Environmental Requirements for Cloud Analysis and Archive (SERCAA), Phillips Laboratory, Hanscom Air Force Base, *Scientific Report No. 2*, PL-TR-94-2114, 100 pp.

Hapke, B. 1993. *Theory of Reflectance and Emittance Spectroscopy*. Cambridge: Cambridge University Press.

Key, J. 1999. *The Cloud and Surface Parameter Retrieval (CASPR) System for Polar AVHRR*. Cooperative Institute for Meteorological Satellite Studies, University of Wisconsin, Madison.

Key, J. and A. J. Schweiger. 1998. Tools for Atmospheric Radiative Transfer: Streamer and FluxNet. *Computers and Geosciences* 24(5):443-451.

Key, J., J. Collins, Chuck Fowler, and R. S. Stone. 1997. High-latitude Surface Temperature Estimates from Thermal Satellite Data. *Remote Sensing of the Environment* 61:302-309.

Kidwell, K. D. 1995. *NOAA Polar Orbiter Data User's Guide*. U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, NESDIS.

Maslanik, J. A., J. Key, Chuck Fowler, T. Nguyen. 2000. AVHRR-derived Regional Cloud and Surface Conditions During SHEBA and FIRE-ACE. *Journal of Geophysical Research*, in press.

Maslanik, J., Chuck Fowler, J. Key, Ted Scambos, T. Hutchinson, and W. Emery. 1998. AVHRR-based Polar Pathfinder Products for Modeling Applications. *Annals of Glaciology* 25:388-392.

Meier, W. N., J. A. Maslanik, J. R. Key, and Chuck Fowler. 1997. Multiparameter AVHRR-derived Products for Arctic Climate Studies. *Earth Interactions* 1.

Rao, C. R. N. 1993. Nonlinearity Corrections for the Thermal Infrared Channels of the AVHRR: Assessment and Recommendations. *NOAA Technical Report, NESDIS* 69:31.

Rao, C. R. N. and J. Chen. 1994. Post-launch Calibration of the Visible and Near-infrared Channels of the Advanced Very High-Resolution Radiometer on NOAA-7, -9, and -11 Spacecraft. *NOAA Technical Report NESDIS* 78:22.

Rao, C. R. N. and J. Chen. 1999. *Revised post-launch calibration of channels 1 and 2 of the Advanced Very High Resolution Radiometer on board the NOAA-14 spacecraft*. http://noaasis.noaa.gov/NOAASIS/ml/aboutn14vis.html.

Rosborough, G. W., D. G. Baldwin, and W. J. Emery. 1994. Precise AVHRR Image Navigation. *IEEE Transactions in Geosciences and Remote Sensing* 32(3):644-657.

Saunders, R. W. and K. T. Kriebel. 1988. An Improved Method for Detecting Clear Sky and Cloudy Radiances from AVHRR Data. *International Journal* of Remote Sensing 9(1):123-150.

Schweiger, A., Chuck Fowler, J. Key, J. Maslanik, J. Francis, Richard Armstrong, Mary Joe Brodzik, Ted Scambos, Terry Haran, M. Ortmeyer, S. Khalsa, D. Rothrock, and Ron Weaver. 1999. P-Cube: A Multisensor Data Set for Polar Climate Research. *Proceedings on the 5th Conference on Polar Meteorology and Oceanography, American Meteorological Society*, Dallas, TX, 15-20 Jan., 136-141.

Schweiger, A. J. and J. R. Key. 1992. Arctic Cloudiness: Comparison of ISCCP-C2 and Nimbus-7 Satellite-derived Cloud Products with a Surface-based Cloud Climatology. *Journal of Climate* 5(12):1514-1527.

Setvak, Martin and Charles A. Doswell III. 1991. The AVHRR Channel 3 Cloud Top Reflectivity of Convective Storms. *American Meteorological Society* 119(3):841-847.

Stowe, L.L., E.P. McClain, R. Carey, P. Pellegrino, and G.G. Gutman. 1991. Global Distribution of Cloud Cover Derived from NOAA/AVHRR Operational Satellite Data. *Advances in Space Research* 11(3): 51- 54.

Stroeve, Julienne C., J. E. Box, Chuck Fowler, Terry Haran, and Jeffrey Key. March 2001. Intercomparison Between in Situ and AVHRR Polar Pathfinder-derived Surface Albedo Over Greenland. *Remote Sensing of the Environment* 75(3):360-374.

Stroeve, Julienne C. 2002. Assessment of Greenland Albedo Variability from the AVHRR Polar Pathfinder Data Set. *Journal of Geophysical Research* 106(D24):33,989-34,005.

Suttles, J. T., R. N. Green, P. Minnis, G .L. Smith, W. F. Staylor, B. A. Wielicki, I. J. Walker, D. F. Young, V. R. Taylor, and L. L. Stowe. 1988. Angular Radiation Models for Earth-Atmosphere System. *Shortwave Radiation, NASA Reference Publication* 1(1184):144.

Walton, C. C., J. T. Sullivan, C. R. N. Rao, and M. P. Weinreb. 1998. Corrections for Detector Nonlinearities and Calibration Inconsistencies of the Infrared Channels of the Advanced Very High-Resolution Radiometer. *Journal of Geophysical Research* 103(C2):3323-3337.

Warren, Stephen G. 1982. Optical Properties of Snow. *Reviews of Geophysics and Space Physics* 20: 67.

Yamanouchi, T., K. Suziki, and S. Kawaguchi. 1987. Detection o*f Clouds in Antarctica from Infrared Multispectral Data of AVHRR.* Journal of Meteorological Society Japan 65(6):949-961.

6 DOCUMENT INFORMATION

6.1 Acronyms and Abbreviations

The following acronyms and abbreviations are used in this document.

Table 19. Acronyms and Abbreviations

APP	AVHRR Polar Pathfinder
APT	Automatic Picture Transmission
ARF	Anisotropic Reflectance Factor
ASCII	American Standard Code for Information Interchange
AVHRR	Advanced Very High Resolution Radiometer
AWS	Automatic Weather Station
BHR	Bihemispherical Reflectance
BRDF	Bidirectional Reflectance Distribution Function
BRF	Bidirectional Reflectance Factor
CASPR	Cloud and Surface Parameter Retrieval
CCAR	Colorado Center for Astrodynamics Research
CIRES	Cooperative Institute for Research in Environmental Sciences
CLAVR	Clouds from AVHRR
DAAC	Distributed Active Archive Center
DEC	Digital Equipment Corporation
DHR	Directional Hemispherical Reflectance
DN	Data (or Digital) Number
EASE-Grid	Equal Area Scalable Earth-Grid
EOS	Earth Observing System
ERBE	Earth Radiation Budget Experiment
FTP	file transfer protocol
GAC	Global Area Coverage
GISMO	Graphical Interface for Subsetting, Mapping, and Ordering
GOES	Geostationary Operational Environmental Satellite
GSFC	Goddard Space Flight Center
HDRF	Hemispherical-Directional Reflectance Factor
HRPT	High Resolution Picture Transmission
IABP	International Arctic Buoy Programme
IFOV	Instantaneous Field of View
JPL	Jet Propulsion Laboratory
LAC	Local Area Coverage
MCC	Maximum Cross Correlation
NASA	National Aeronautics and Space Administration
NAVSPASUR	Naval Space Surveillance Center
NCAR	National Center for Atmospheric Research

NSIDC	National Snow and Data Center
NOAA	National Oceanic and Atmospheric Administration
POD	Polar Orbiter Data
POES	Polar Orbiting Environmental Satellites
RMS	Root Mean Square
RMSE	Root Mean Square Error
SAA	Satellite Active Archive
SERCAA	Support of Environmental Requirements for Cloud Analysis and Archives
SHEBA	Surface Heat Balance of the Arctic
SMMR	Scanning Multichannel Microwave Radiometer
SSM/I	Special Sensor Microwave/Imager
TOA	Top of Atmosphere
TIROS	Television and Infrared Observation Satellite
TOVS	TIROS Operational Vertical Sounder
UTC	Universal Coordinated Time

6.2 Publication Date

2000

6.3 Date Last Updated

18 March 2021

APPENDIX A ANISOTROPIC REFLECTANCE VALUES AND NORMALIZING FACTORS FOR ALBMN

```
* anisdata.h     */
static float rmatrx[3][10][7][8] =
{
  {
    {
      {
      1.08065033, 1.08065033, 1.08065033, 1.08065033,
      1.08065033, 1.08065033, 1.08065033, 1.08065033
      },
      {
      2.15055275, 1.77242291, 1.25359344, 0.99564439,
      0.89305103, 0.87057817, 0.86764693, 0.87448651
      },
      {
      1.74604166, 1.35618687, 0.97414863, 0.85592198,
      0.83051789, 0.84517407, 0.85299075, 0.85983026
      },
      {
      1.02886510, 0.94092792, 0.82367831, 0.78654927,
      0.81586170, 0.85983026, 0.88914269, 0.91552383
      },
      {
      0.95949244, 0.89695930, 0.85983026, 0.83051789,
      0.85592198, 0.91845506, 0.95949244, 1.00346100
      },
      {
      1.11777937, 1.09921479, 1.05817747, 1.01811719,
      1.05524635, 1.06990254, 1.13243556, 1.19789994
      },
      {
      1.41578865, 1.39722419, 1.33469105, 1.23502898,
      1.26824975, 1.29756212, 1.36009514, 1.39722419
      }
    },
    {
      {
      0.81155437, 0.81155437, 0.81155437, 0.81155437,
      0.81155437, 0.81155437, 0.81155437, 0.81155437
      },
      {
      1.46373117, 1.33075356, 0.96897638, 0.77733219,
      0.76071000, 0.77733219, 0.78808773, 0.81546551
      },
      {
      2.23128581, 1.60648656, 0.90053201, 0.72942120,
      0.74995446, 0.82915437, 0.86337656, 0.89368761
      },
      {
```

```
   1.71599746, 1.29946482, 0.83893210, 0.71964341,
   0.77733219, 0.89368761, 0.96213192, 0.98266524
   },
   {
   1.36497581, 1.13226497, 0.88684314, 0.77048779,
   0.86630988, 0.98950970, 1.08826506, 1.11564279
   },
   {
   1.48035336, 1.30630922, 1.11857617, 1.01004303,
   0.92790979, 1.13617611, 1.31999815, 1.35813141
   },
   {
   1.80790854, 1.68568647, 1.47057569, 1.21830940,
   1.12248731, 1.41582024, 1.52533114, 1.58986437
   }
  },
  {
   {
   0.66493076, 0.66493076, 0.66493076, 0.66493076,
   0.66493076, 0.66493076, 0.66493076, 0.66493076
   },
   {
   0.92503601, 0.90352356, 0.73337954, 0.63363987,
   0.68057621, 0.71773410, 0.74902493, 0.74902493
   },
   {
   1.81291425, 1.36799729, 0.75880337, 0.63950694,
   0.69622165, 0.79302770, 0.85658729, 0.84974241
   },
   {
   2.83964539, 1.56258726, 0.79693907, 0.67764264,
   0.72751248, 0.88396680, 0.97197235, 1.01597512
   },
   {
   2.49055672, 1.44329095, 0.87809974, 0.75586981,
   0.82822990, 1.00032973, 1.12940454, 1.15385044
   },
   {
   2.14244604, 1.56845427, 1.17927432, 0.90645707,
   0.94361496, 1.21056509, 1.39928806, 1.44622445
   },
   {
   2.44655418, 2.00457072, 1.54009700, 1.32692802,
   1.18783045, 1.53716350, 1.72881997, 1.75326598
   }
  },
  {
   {
   0.58392209, 0.58392209, 0.58392209, 0.58392209,
   0.58392209, 0.58392209, 0.58392209, 0.58392209
   },
   {
   0.67879719, 0.70716190, 0.59859347, 0.56925070,
   0.60935253, 0.68662196, 0.71889907, 0.69835907
   },
   {
   1.22261739, 0.99080914, 0.64162964, 0.57511926,
   0.62989253, 0.75900090, 0.81573033, 0.82159895
   },
   {
   2.98123050, 1.56886268, 0.74726379, 0.64162964,
   0.68173146, 0.84702939, 0.95657587, 1.00254631
   },
   {
   3.98377681, 1.70677388, 0.85876650, 0.70129335,
   0.76975995, 0.98787487, 1.14828229, 1.19425273
   },
   {
   3.66002750, 1.98161829, 1.20305562, 0.88419694,
   0.93897021, 1.25391638, 1.48376846, 1.53169513
   },
   {
   3.75490260, 2.38557124, 1.62070167, 1.25391638,
   1.28277016, 1.62950444, 1.81534243, 1.84957576
   }
  },
  {
   {
   0.50470167, 0.50470167, 0.50470167, 0.50470167,
   0.50470167, 0.50470167, 0.50470167, 0.50470167
   },
   {
   0.56827843, 0.53893530, 0.53111047, 0.50470167,
   0.52915424, 0.61522740, 0.63087708, 0.63087708
   },
   {
   0.88322788, 0.81378251, 0.54871637, 0.51448268,
   0.56143171, 0.67000121, 0.73944658, 0.76389921
   },
   {
   2.61055946, 1.41042602, 0.71205968, 0.59859961,
   0.60838068, 0.79911095, 0.91257101, 0.92235208
   },
   {
   5.07733774, 1.91023707, 0.87051255, 0.66022021,
   0.71499401, 0.96245432, 1.09058595, 1.16296566
   },
   {
   6.45059586, 2.20660257, 1.35076165, 0.87540305,
   0.84605998, 1.25392926, 1.47889316, 1.52682030
   },
   {
   7.11864090, 3.05364060, 1.78895223, 1.30658388,
   1.26631856, 1.63538992, 1.83101070, 1.90241230
   }
  },
  {
   {
   0.45716840, 0.45716840, 0.45716840, 0.45716840,
   0.45716840, 0.45716840, 0.45716840, 0.45716840
   },
   {
   0.52275789, 0.52471578, 0.48751575, 0.46402103,
   0.47968417, 0.54429471, 0.59128416, 0.62358946
   },
   {
   0.78609467, 0.70582104, 0.55310524, 0.47087365,
   0.51590520, 0.61967367, 0.67155784, 0.72148418
   },
   {
   2.18011570, 1.26382101, 0.66274732, 0.54625261,
   0.58345258, 0.75378942, 0.83406311, 0.88105255
   },
   {
   5.35679960, 1.80811572, 0.87713677, 0.66764206,
   0.72148418, 0.88790518, 1.08663154, 1.16396844
   },
   {
   9.64850426, 2.58148408, 1.34409475, 0.75770521,
   0.84923679, 1.19627357, 1.46450520, 1.60057890
   },
   {
   10.55794716, 3.41163135, 1.76699984, 1.26235259,
   1.13275945, 1.54673684, 1.89719987, 1.99019992
   }
  },
  {
   {
   0.46849415, 0.46849415, 0.46849415, 0.46849415,
   0.46849415, 0.46849415, 0.46849415, 0.46849415
   },
   {
   0.44070211, 0.50025648, 0.41985810, 0.44632670,
   0.46932128, 0.46849415, 0.50918961, 0.50720447
   },
   {
   0.83344316, 0.66601604, 0.51315987, 0.44467241,
   0.50422674, 0.58090299, 0.62829828, 0.65112746
   },
   {
   1.58513796, 1.13748789, 0.68189722, 0.50025648,
   0.53102618, 0.72259265, 0.85063452, 0.81093162
   },
   {
   4.46955347, 2.19159961, 0.89927053, 0.63822401,
   0.65311259, 0.93500316, 1.08754170, 1.06602275
   },
   {
   10.19868088, 2.47746062, 1.64072216, 0.71499997,
   0.97387892, 1.12160683, 1.50573230, 1.58315289
   },
   {
   11.41401672, 3.26357794, 2.45215011, 1.49409926,
   1.36535168, 1.61987805, 2.11814952, 2.09432769
   }
  },
  {
   {
   0.43962979, 0.43962979, 0.43962979, 0.43962979,
   0.43962979, 0.43962979, 0.43962979, 0.43962979
   },
   {
   0.54989105, 0.52974218, 0.49456441, 0.45649880,
   0.45908639, 0.48954341, 0.52339286, 0.54457819
   },
   {
   0.77370900, 0.67925209, 0.57311010, 0.48995844,
   0.45412293, 0.56838453, 0.63759172, 0.65884817
   },
   {
   1.52447331, 1.00452995, 0.70007092, 0.55077189,
   0.54997909, 0.71190661, 0.81371242, 0.83358377
   },
   {
   4.38874960, 1.63727224, 0.96541911, 0.66414428,
   0.64403713, 0.88550133, 1.02214873, 1.06196880
   },
   {
   9.12249279, 2.42335629, 1.69647360, 0.91342282,
   0.97580683, 1.15146863, 1.51774466, 1.54702151
   },
   {
   12.12200165, 3.18738699, 2.95976520, 1.67166591,
   1.54238105, 1.68606031, 2.29356861, 2.18667769
   }
  },
  {
   {
   0.42262459, 0.42262459, 0.42262459, 0.42262459,
   0.42262459, 0.42262459, 0.42262459, 0.42262459
   },
   {
   0.53466427, 0.51878291, 0.48404339, 0.43776861,
   0.43906945, 0.47095078, 0.49810594, 0.51436192
   },
   {
   0.73975128, 0.66760683, 0.56760722, 0.47545472,
   0.43650633, 0.55210394, 0.60280263, 0.62467003
   },
   {
   1.38743651, 0.97146976, 0.69614089, 0.56539828,
   0.54451340, 0.69903445, 0.78498709, 0.79895437
   },
   {
   4.04669476, 1.65192425, 0.95911962, 0.69305158,
   0.66798925, 0.87439930, 0.99056542, 1.02997553
   },
   {
   8.46174145, 2.40327024, 1.76190066, 1.06085527,
   0.99044091, 1.18042445, 1.54147899, 1.53480828
   },
   {
   12.76416397, 3.15700436, 3.37077498, 1.82092500,
   1.69011772, 1.75085592, 2.44655848, 2.27566123
   }
  },
  {
   {
     0.41042590, 0.41042590, 0.41042590, 0.41042590,
     0.41042590, 0.41042590, 0.41042590, 0.41042590
     },
     {
     0.52422702, 0.51168734, 0.47718802, 0.42424628,
     0.42452872, 0.45764092, 0.47956112, 0.49195078
     },
     {
     0.71506780, 0.66042876, 0.56499910, 0.46540797,
     0.42386493, 0.54088354, 0.57703274, 0.59945363
     },
     {
     1.28285873, 0.94821990, 0.69518214, 0.57877827,
     0.54186308, 0.69097918, 0.76460195, 0.77392119
     },
     {
     3.78747392, 1.66870224, 0.95709556, 0.71817625,
     0.68909854, 0.86829585, 0.96854836, 1.00775528
     },
     {
     7.96300697, 2.39478087, 1.81936705, 1.18130481,
     1.00514841, 1.20710266, 1.56513059, 1.52987504
     },
     {
     12.39235210, 3.14137292, 3.60768747, 2.07689571,
     1.75645888, 1.85096824, 2.55765390, 2.33108616
     }
   }
  },
  {
   {
     {
     1.08353364, 1.08353364, 1.08353364, 1.08353364,
     1.08353364, 1.08353364, 1.08353364, 1.08353364
     },
     {
     0.98387587, 0.95554179, 0.98680699, 1.01318705,
     1.04347515, 1.09134996, 1.13433945, 1.16951275
     },
     {
     0.90864402, 0.90864402, 0.93502402, 0.98387587,
     1.01709509, 1.06496990, 1.10502839, 1.10112035
     },
     {
     0.87249368, 0.88714921, 0.91059810, 0.93502402,
     1.01318705, 1.04152095, 1.06790102, 1.07962549
     },
     {
     0.91059810, 0.88519520, 0.89496553, 0.95163363,
     1.00048554, 1.05129135, 1.04836035, 1.08353364
     },
     {
     0.95065659, 0.93013883, 0.93209291, 0.95065659,
     0.99657732, 1.03175068, 1.03468180, 1.06692398
     },
     {
     1.04347515, 0.99657732, 0.97899067, 0.97215146,
     0.98876107, 1.03858995, 1.05813062, 1.08646464
     }
   },
   {
     {
     0.94294357, 0.94294357, 0.94294357, 0.94294357,
     0.94294357, 0.94294357, 0.94294357, 0.94294357
     },
     {
     0.87161207, 0.83350348, 0.89310926, 0.91851497,
     0.98691499, 1.04456651, 1.10808074, 1.15791512
     },
     {
     0.88236064, 0.82959491, 0.85402352, 0.90483499,
     0.95173782, 1.08853805, 1.19895518, 1.25758386
     },
     {
     0.83350348, 0.84327489, 0.85988635, 0.90874350,
     0.97323501, 1.11101234, 1.19504654, 1.23901808
     },
     {
     0.88920069, 0.86965781, 0.89310926, 0.89310926,
     1.00450361, 1.11687505, 1.20579517, 1.23510945
     },
     {
     1.02697790, 0.99961787, 0.98398358, 0.95173782,
     1.01818359, 1.11687505, 1.23315525, 1.25953805
     },
     {
     1.19504654, 1.16182375, 1.09537804, 1.01818359,
     1.05678082, 1.17257226, 1.24097240, 1.31034946
     }
   },
   {
     {
     0.88637948, 0.88637948, 0.88637948, 0.88637948,
     0.88637948, 0.88637948, 0.88637948, 0.88637948
     },
     {
     0.81894821, 0.76813042, 0.82383454, 0.88344771,
     0.94110638, 0.97921968, 1.03199196, 1.05544639
     },
     {
     0.74956238, 0.78083485, 0.81406188, 0.89615214,
     0.94599265, 1.08183253, 1.18444538, 1.21180880
     },
     {
     0.84728891, 0.81113011, 0.83556169, 0.89028853,
     0.94403815, 1.12092304, 1.26849020, 1.36230755
     },
     {
     0.90494752, 0.86878872, 0.88051593, 0.88833404,
     0.99778771, 1.18542266, 1.31246710, 1.35448945
     },
     {
     1.07303715, 1.02417386, 0.98703784, 0.95087898,
     1.04176474, 1.23330855, 1.35546684, 1.40726185
```

```
      },
      {
      1.27337646, 1.19226348, 1.15219557, 1.00071955,
      1.10015631, 1.31637621, 1.41996634, 1.43657982
      }
    },
    {
      {
      0.82926643, 0.82926643, 0.82926643, 0.82926643,
      0.82926643, 0.82926643, 0.82926643, 0.82926643
      },
      {
      0.74633980, 0.73073006, 0.75902265, 0.80780303,
      0.88292480, 0.95024174, 0.99414408, 1.01365626
      },
      {
      0.72292519, 0.74829096, 0.76097387, 0.79316890,
      0.91121745, 1.03414404, 1.14536333, 1.12877786
      },
      {
      0.87902242, 0.80585182, 0.78731531, 0.83609563,
      0.90048575, 1.13658285, 1.30633855, 1.40487492
      },
      {
      1.02341235, 0.86633950, 0.87219316, 0.84585172,
      0.98536360, 1.21853375, 1.41170406, 1.54731357
      },
      {
      1.16194856, 1.12682664, 1.04877806, 0.95414418,
      1.03219283, 1.35511887, 1.48780155, 1.59609389
      },
      {
      1.46146011, 1.36780179, 1.23219228, 1.13967216,
      1.22520053, 1.51121604, 1.61560607, 1.64584994
      }
    },
    {
      {
      0.75580239, 0.75580239, 0.75580239, 0.75580239,
      0.75580239, 0.75580239, 0.75580239, 0.75580239
      },
      {
      0.71278131, 0.70300382, 0.70789254, 0.75971341,
      0.83988911, 0.90735400, 0.92202026, 0.93179780
      },
      {
      0.73722512, 0.73429185, 0.73429185, 0.75580239,
      0.85846639, 1.01588452, 1.09997118, 1.09214926
      },
      {
      0.82815605, 0.79980123, 0.80860102, 0.79980123,
      0.86139965, 1.13810349, 1.27694428, 1.30921006
      },
      {
      1.05890560, 0.90735400, 0.86531067, 0.88095468,
      1.00317371, 1.23978972, 1.40698528, 1.59569144
      },
      {
      1.32974291, 1.18503559, 1.09801567, 0.94157535,
      1.03739500, 1.52529335, 1.63089061, 1.71008849
      },
      {
      1.70128870, 1.51551580, 1.32289875, 1.18878365,
      1.29324019, 1.66217875, 1.77266455, 1.81861901
      }
    },
    {
      {
      0.71193153, 0.71193153, 0.71193153, 0.71193153,
      0.71193153, 0.71193153, 0.71193153, 0.71193153
      },
      {
      0.68552750, 0.67868197, 0.68063784, 0.72073287,
      0.77647477, 0.82439327, 0.91240674, 0.88893652
      },
      {
      0.71388739, 0.65325588, 0.70019639, 0.73051214,
      0.79407746, 0.94663423, 1.07474279, 1.07474279
      },
      {
      0.82732701, 0.77843064, 0.75202656, 0.78429818,
      0.84981936, 1.07180905, 1.23414505, 1.36225355
      },
      {
      1.24392426, 0.93392116, 0.87817925, 0.86057657,
      0.98575133, 1.21947610, 1.45320094, 1.63998508
      },
      {
      1.57250810, 1.35443020, 1.15688872, 0.96619278,
      1.03855956, 1.52947927, 1.72213113, 2.14068413
      },
      {
      2.11916971, 1.83459282, 1.51187658, 1.27538097,
      1.34791064, 1.72995448, 1.92065048, 2.15828681
      }
    },
    {
      {
      0.63841540, 0.63841540, 0.63841540, 0.63841540,
      0.63841540, 0.63841540, 0.63841540, 0.63841540
      },
      {
      0.64609557, 0.68161643, 0.65377575, 0.59137428,
      0.73633778, 0.79681921, 0.76993859, 0.80833954
      },
      {
      0.74401796, 0.76225841, 0.74401796, 0.69961691,
      0.81073958, 0.90962195, 0.98690385, 0.95186299
      },
      {
      0.85154057, 0.85154057, 0.83586019, 0.71905738,
      0.88514137, 1.02242470, 1.20386910, 1.09538639
      },
      {
```

```
   1.63971961, 1.04258525, 0.97906363, 0.81601971,
   0.87458110, 1.32867205, 1.32675195, 1.49917221
   },
   {
   2.23685408, 1.45923519, 1.17122829, 0.94826287,
   0.98978388, 1.48899591, 1.71652138, 1.87273645
   },
   {
   3.23030090, 1.93499148, 1.63519609, 1.33540082,
   1.37692177, 1.76068246, 2.03236914, 2.49670315
   }
  },
  {
   {
   0.61749113, 0.61749113, 0.61749113, 0.61749113,
   0.61749113, 0.61749113, 0.61749113, 0.61749113
   },
   {
   0.68142474, 0.65988243, 0.65023404, 0.64567739,
   0.67042756, 0.70827103, 0.72796732, 0.75072420
   },
   {
   0.76763827, 0.74312127, 0.71612966, 0.67535841,
   0.71679646, 0.80105036, 0.86788672, 0.89861208
   },
   {
   0.89334583, 0.84752315, 0.79620296, 0.72967130,
   0.79375172, 0.97712481, 1.06150401, 1.09999776
   },
   {
   1.22825658, 1.09088814, 0.97322929, 0.85152465,
   0.92971027, 1.27702403, 1.36937714, 1.43923128
   },
   {
   2.50437689, 1.58209777, 1.30693996, 1.06499624,
   1.10580826, 1.52595699, 1.75730062, 2.04239798
   },
   {
   4.29087830, 2.07955432, 1.79076934, 1.42712057,
   1.44294894, 1.83958220, 2.18964911, 2.86459112
   }
  },
  {
   {
   0.57617033, 0.57617033, 0.57617033, 0.57617033,
   0.57617033, 0.57617033, 0.57617033, 0.57617033
   },
   {
   0.66547453, 0.63941568, 0.62311703, 0.61068195,
   0.62579876, 0.66525811, 0.68418938, 0.70723808
   },
   {
   0.76470119, 0.73896682, 0.70165080, 0.64768243,
   0.68012679, 0.76386398, 0.81399906, 0.85118723
   },
   {
   0.90691227, 0.85078275, 0.79415476, 0.71478540,
   0.77692157, 0.94632220, 1.01805675, 1.03726661
   },
   {
   1.27255642, 1.13022745, 0.98518276, 0.87712300,
   0.95262259, 1.23753774, 1.32505798, 1.37502182
   },
   {
   2.71062303, 1.65964508, 1.39610517, 1.14140093,
   1.17945528, 1.52655077, 1.75838113, 2.13320804
   },
   {
   5.14310551, 2.16571283, 1.89246655, 1.47962248,
   1.47240102, 1.87246120, 2.28484893, 3.13258123
   }
  },
  {
   {
   0.53879744, 0.53879744, 0.53879744, 0.53879744,
   0.53879744, 0.53879744, 0.53879744, 0.53879744
   },
   {
   0.64999098, 0.62005961, 0.59800631, 0.57873726,
   0.58542562, 0.62619859, 0.64441812, 0.66766614
   },
   {
   0.76031774, 0.73358423, 0.68736273, 0.62202603,
   0.64655787, 0.72964513, 0.76511067, 0.80784041
   },
   {
   0.91656709, 0.85159236, 0.79047537, 0.70011145,
   0.76040554, 0.91722935, 0.97777694, 0.98014116
   },
   {
   1.30811310, 1.16180217, 0.99324560, 0.89733088,
   0.97031021, 1.20018494, 1.28328621, 1.31580245
   },
   {
   2.88383317, 1.72324276, 1.47045314, 1.20523965,
   1.24080086, 1.52342546, 1.75512660, 2.20723009
   },
   {
   5.62937355, 2.22819257, 2.03402424, 1.58947074,
   1.57831526, 1.91478431, 2.35891843, 3.58372641
   }
  }
 },
 {
  {
   {
   1.05506957, 1.05506957, 1.05506957, 1.05506957,
   1.05506957, 1.05506957, 1.05506957, 1.05506957
   },
   {
   1.05857182, 1.11124825, 0.96654630, 1.07797289,
   1.06672049, 1.08333778, 0.98989403, 1.03323054
   },
   {
```

```
    1.05026495, 1.08609688, 0.98041075, 1.06020474,
    1.05299890, 1.06762385, 0.99634749, 1.03300869
    },
    {
    1.03868842, 1.05104566, 0.99973249, 1.03544295,
    1.03387630, 1.04572475, 1.00534105, 1.03269970
    },
    {
    1.02179277, 1.00724816, 1.01886642, 1.00325966,
    1.00759041, 1.01636970, 1.01298606, 1.02804542
    },
    {
    0.94540453, 0.94893414, 0.94181645, 0.92076230,
    0.92054677, 0.92726398, 0.92904413, 0.92963237
    },
    {
    0.76746947, 0.76721472, 0.75600398, 0.72617435,
    0.73096538, 0.74065691, 0.74600756, 0.74834836
    }
  },
  {
    {
    1.01221609, 1.01221609, 1.01221609, 1.01221609,
    1.01221609, 1.01221609, 1.01221609, 1.01221609
    },
    {
    1.03091943, 1.06099761, 0.97447371, 1.04560316,
    1.03878045, 1.07374787, 0.98616499, 1.01105642
    },
    {
    1.02158749, 1.02246237, 1.02024627, 1.04471636,
    1.04070640, 1.14298832, 1.01395142, 0.99642718
    },
    {
    1.02996016, 1.02292955, 1.01726246, 1.02558339,
    1.01737571, 1.08275986, 1.01509655, 1.00632322
    },
    {
    1.03838468, 1.02170265, 1.01152706, 1.00048649,
    0.98761290, 1.01042068, 1.01384997, 1.01555884
    },
    {
    0.97974622, 0.98893923, 0.95768917, 0.92645282,
    0.92267466, 0.94329333, 0.94896442, 0.96050197
    },
    {
    0.88038772, 0.84198183, 0.79656422, 0.75724995,
    0.73666042, 0.76798952, 0.78260946, 0.81823730
    }
  },
  {
    {
    0.99386787, 0.99386787, 0.99386787, 0.99386787,
    0.99386787, 0.99386787, 0.99386787, 0.99386787
    },
    {
    1.01345682, 1.02829015, 0.97366416, 1.01577604,
    1.01159894, 1.03842461, 0.97992879, 1.00059819
    },
    {
    1.02692294, 1.01481795, 1.01231956, 1.02294850,
    1.01473379, 1.08503807, 0.99966234, 0.99737817
    },
    {
    1.08861959, 1.04128325, 1.04247272, 1.02299023,
    1.00645518, 1.04271245, 0.99262100, 1.03008819
    },
    {
    1.08411157, 1.05670774, 1.03626096, 0.97653723,
    0.96127677, 1.01090264, 1.02010274, 1.02471173
    },
    {
    1.07867289, 1.04925430, 1.00089705, 0.93577754,
    0.93853664, 0.95597607, 0.96613359, 0.98620564
    },
    {
    1.03433800, 0.97064126, 0.87486345, 0.79197687,
    0.76236504, 0.78796625, 0.83324099, 0.85597646
    }
  },
  {
    {
    0.96636963, 0.96636963, 0.96636963, 0.96636963,
    0.96636963, 0.96636963, 0.96636963, 0.96636963
    },
    {
    0.98786622, 0.99177474, 0.95171291, 0.97614086,
    0.97711796, 0.99177474, 0.95366710, 0.98591202
    },
    {
    0.99959165, 0.98688912, 0.98688912, 0.98395777,
    0.97223234, 1.02792811, 0.98591202, 0.97320950
    },
    {
    1.09632635, 1.04258490, 1.03379083, 0.98395777,
    0.96539253, 0.99763745, 0.97614086, 1.02304244
    },
    {
    1.13638818, 1.08069253, 1.04551625, 0.91555953,
    0.95659846, 0.97125524, 1.01815689, 1.03379083
    },
    {
    1.17742705, 1.12563992, 1.07580686, 0.97809505,
    0.94682729, 0.98004931, 0.98786622, 0.98395777
    },
    {
    1.26341355, 1.19012976, 1.08264673, 1.02401960,
    1.00203454, 1.04453909, 0.97907221, 0.92142224
    }
  },
  {
    {
    0.95177984, 0.95177984, 0.95177984, 0.95177984,
    0.95177984, 0.95177984, 0.95177984, 0.95177984
```

```
      },
      {
      0.97816384, 0.95764291, 0.97425514, 0.95275700,
      0.95959735, 0.97132355, 0.96546042, 0.98793572
      },
      {
      1.00552511, 0.99086732, 0.97620946, 0.96839201,
      0.95177984, 0.97523230, 0.98109543, 0.97816384
      },
      {
      1.08272278, 1.05145288, 1.03093195, 0.94298512,
      0.92735016, 0.99282163, 1.00357068, 1.01236534
      },
      {
      1.22148335, 1.12474191, 1.06122470, 0.93223608,
      0.93223608, 0.98109543, 1.01725125, 1.05829322
      },
      {
      1.29477227, 1.21171153, 1.12278748, 0.97132355,
      0.93809921, 0.99477601, 1.00552511, 1.02311444
      },
      {
      1.46382558, 1.32701945, 1.15796626, 1.02604604,
      0.97083497, 0.92637295, 0.99477601, 0.99966198
      }
    },
    {
      {
      0.89720362, 0.89720362, 0.89720362, 0.89720362,
      0.89720362, 0.89720362, 0.89720362, 0.89720362
      },
      {
      0.94216150, 0.93043333, 0.93727481, 0.91870522,
      0.91381848, 0.92847866, 0.93825215, 0.93629742
      },
      {
      0.99102879, 0.96854985, 0.94900292, 0.93336540,
      0.92261457, 0.95291233, 0.94704825, 0.95291233
      },
      {
      1.10733294, 1.05357897, 1.00666630, 0.94313884,
      0.94607091, 0.96757251, 0.98418736, 0.98614210
      },
      {
      1.24416137, 1.14544952, 1.07117116, 0.96073109,
      0.95388967, 0.98418736, 1.01350784, 1.03794146
      },
      {
      1.46210957, 1.32430375, 1.18063390, 0.96757251,
      0.97050452, 1.02328122, 1.04576015, 1.06042039
      },
      {
      1.70058203, 1.50902224, 1.28227794, 1.07947862,
      1.03549814, 1.00959837, 1.07312596, 1.04185081
      }
    },
    {
```

```
    {
    0.86022455, 0.86022455, 0.86022455, 0.86022455,
    0.86022455, 0.86022455, 0.86022455, 0.86022455
    },
    {
    0.90616834, 0.91203350, 0.90128070, 0.87195486,
    0.87195486, 0.88075262, 0.88270766, 0.88368517
    },
    {
    0.94722450, 0.95895481, 0.92376381, 0.89248294,
    0.88857281, 0.91007847, 0.91692114, 0.93060654
    },
    {
    1.10656154, 1.05670762, 1.00196612, 0.91887617,
    0.92571890, 0.93940431, 0.95113462, 0.97557282
    },
    {
    1.26785362, 1.18769646, 1.09776378, 0.94233686,
    0.93647170, 0.98534811, 1.02347171, 1.01565146
    },
    {
    1.71751654, 1.47997725, 1.25416827, 0.96873009,
    0.98143798, 1.04888737, 1.10069644, 1.12806726
    },
    {
    2.21801066, 1.77421308, 1.37733674, 1.13230312,
    1.09792674, 1.13295484, 1.16325819, 1.17792106
    }
   },
   {
    {
    0.81249756, 0.81249756, 0.81249756, 0.81249756,
    0.81249756, 0.81249756, 0.81249756, 0.81249756
    },
    {
    0.85942882, 0.87996125, 0.86040658, 0.82716358,
    0.81738621, 0.83400768, 0.83205223, 0.83400768
    },
    {
    0.93275893, 0.92493707, 0.89658272, 0.83596319,
    0.84182960, 0.87800574, 0.86725068, 0.88876086
    },
    {
    1.13612771, 1.07355273, 0.96991283, 0.88582766,
    0.88387215, 0.90244913, 0.94351399, 0.94351399
    },
    {
    1.35122943, 1.28669894, 1.12146175, 0.91124874,
    0.91320419, 0.96697962, 1.01586640, 1.01684415
    },
    {
    2.04835415, 1.69832516, 1.34145200, 0.93080348,
    0.96306866, 1.10190701, 1.15666020, 1.18501449
    },
    {
    3.02217793, 2.21163583, 1.55459821, 1.18468857,
    1.14280891, 1.20652461, 1.28865445, 1.33069694
    }
  },
  {
    {
    0.73465902, 0.73465902, 0.73465902, 0.73465902,
    0.73465902, 0.73465902, 0.73465902, 0.73465902
    },
    {
    0.79335350, 0.79041874, 0.77476692, 0.73661548,
    0.74248493, 0.75813681, 0.76302803, 0.76400626
    },
    {
    0.92737252, 0.87748224, 0.82661366, 0.78944051,
    0.75226736, 0.79530996, 0.81096184, 0.84617847
    },
    {
    1.14454198, 1.06726098, 0.94987202, 0.84128731,
    0.80411410, 0.82954842, 0.89606881, 0.90389472
    },
    {
    1.51431715, 1.35486376, 1.12693369, 0.88432992,
    0.87552571, 0.92248130, 1.01345766, 1.00856650
    },
    {
    2.74200964, 2.06506681, 1.54464257, 0.93324196,
    0.92932898, 1.10541236, 1.20910585, 1.23649669
    },
    {
    4.43436670, 3.01787353, 1.99169874, 1.34883130,
    1.23323596, 1.30986476, 1.43997085, 1.50062168
    }
  },
  {
    {
    0.66551805, 0.66551805, 0.66551805, 0.66551805,
    0.66551805, 0.66551805, 0.66551805, 0.66551805
    },
    {
    0.73500592, 0.74479294, 0.72717631, 0.66649669,
    0.67726243, 0.68411338, 0.70270872, 0.71543187
    },
    {
    0.85440767, 0.82015306, 0.76338834, 0.71641058,
    0.73206979, 0.73696333, 0.75555867, 0.82798272
    },
    {
    1.12257230, 1.05406308, 0.90530026, 0.78785586,
    0.77415407, 0.77806884, 0.82798272, 0.85832250
    },
    {
    1.61192381, 1.41618323, 1.11865747, 0.81525958,
    0.84853548, 0.91606599, 1.01393628, 0.98555386
    },
    {
    3.09563756, 2.38314176, 1.64226365, 0.95521408,
    0.94542706, 1.15682685, 1.27524996, 1.28112221
    },
      {
      5.90255737, 3.66817880, 2.31658983, 1.49611056,
      1.35142565, 1.47784150, 1.58941364, 1.66575241
      }
    }
  }
};

static float albmn[3][10] =
{
  {
  1.00051761, 1.00434864, 1.00560844, 1.00906873, 1.01032078,
  1.01504338, 1.00748992, 1.00088763, 0.99531543, 0.99175829
  },
  {
  0.99871516, 1.00025940, 0.99998677, 0.99918854, 0.99875021,
  0.99897498, 0.99636352, 0.99569529, 0.99525517, 0.99399608
  },
  {
  1.00432646, 1.00373149, 1.00351846, 0.99983281, 1.00079358,
  1.00041556, 0.99936169, 0.99813300, 0.99535543, 0.99226213
  }
};
```

APPENDIX B: ATMOSPHERIC AND ANISOTROPIC REFLECTANCE CORRECTION SOURCE CODE DOCUMENT

Procedures in this file are for atmospheric and anisotropic reflectance correction of the AVHRR shortwave channel. They are:

call_narr2bb_toa
call_anis_correct
call_anis_correct_blend
call_atmos_correct
narr2bb_toa
anis_correct
anis_correct_blend
atmos_correct

Compilation: See the file README.C for instructions on compiling this file.

```
#include <assert.h>
#include <errno.h>
#include <stdio.h>
#include <math.h>
#include "globals.h"
#include "anisdata.h"

long call_narr2bb_toa(short argc, void *argv[]);
long call_anis_correct(short argc, void *argv[]);
long call_anis_correct_blend(short argc, void *argv[]);
long call_atmos_correct(short argc, void *argv[]);
float narr2bb_toa(float ch1, float ch2, float solzen, char surftype);
float anis_correct(float broadalb, float solzen, float scan, float relaz, char surftype);
float anis_correct_blend(float ch1, float ch1_pure_ocean, float ch1_pure_ice, float broadalb, float solzen, float scan, float relaz, char surftype);
float atmos_correct(float broadalb, float solzen, float wv, float odhaze, char surftype);

/******************** CALL NARROW TO BROADBAND CONVERSION ******************/

long call_narr2bb_toa(short argc, void *argv[])

/*---------------------------------------------------------------------------
```

This function loops through the input arrays and calls 'surfalb_clear' (below) for each pixel. It is called from an IDL program with the following statement:

ERROR =
call_external(objfln,'call_narr2bb_toa',ncols,nrows,rsurf1,rsurf2,$ solzen,surftype,cmask,badpix,broadalbtoa)

Input (as pointers to the specified type):

ncols, nrows : Dimensions of 'tsurf' and the arrays pointed to by 'array' [short integer]. While these arrays are 2D, they are treated here as 1D. If scalar values are passed in, both of these values must be 1. If a 1D array is passed in, then 'nrows' must be 1.

ch1, ch2 : Channels 1 and 2 albedos at TOA, in the range [0,1] and normalized for solar zenith angle and NOT corrected for anisotropy and the atmosphere. [float]

solzen : Solar zenith angle image (degrees) [float]

surftype : Surface type (global variables LAND, OCEAN, ICE, SNOW; see 'globals.pro' for values). LAND means snow-free land (vegetation cover), OCEAN means open water, ICE means sea ice (with or without snow), and SNOW means snow-covered land. However, corrections are currently done for only land and ice surface. For other surface types the channel 1 reflectance is returned. [char]

cmask : Cloud mask array, cloudy=CLOUDY, clear=CLEAR ('cmask.pro' and 'globals.h') [char]

badpix : Valid pixels array, where invalid=TRUE ('globals.h') [char]

broadalbtoa : The albedo array. The values are unimportant, but this array must be defined upon entry. [float]

Output:

broadalbtoa: Broadband TOA albedo in the range [0,1]. For surface types other than snow/ice or vegetation the clear sky albedo is returned. [fltarr or float]. If badpix=TRUE and/or the solar zenith angle is > DARKZEN, MISSING values are returned. For CLOUDY pixels the input values are unchanged. Results outside of the range [0,1] are reset to 0 or 1.

Additional Variables (from globals.h):

TRUE, FALSE : true and false flags to be used for returning error condition

Called by: narr2bb_toa (an IDL procedure)

Calls: narr2bb_toa (the C function below)

```
------------------------------------------------------------------------*/
{

/* parameters passed in from IDL */

short ncols, nrows;
char *cmask, *surftype, *badpix;
float *ch1, *ch2, *solzen;
float *broadalbtoa;


/* local parameters */

long i;
long ERROR; /* Error flag, returned to IDL */

/* Check that correct number of parameters was passed */

if (argc != 9) {
fprintf(stdout, "argc = %d\r\n",argc);
fprintf(stdout, "ERROR >>> CALL_NARR2BB_TOA: INCORRECT NUMBER OF
ARGUMENTS\r\n");
ERROR = TRUE;
return ERROR;
}

/* Cast passed parameters to local vars */

ncols = *(short *) argv[0];
nrows = *(short *) argv[1];
ch1 = (float *) argv[2];
ch2 = (float *) argv[3];
solzen = (float *) argv[4];
surftype = (char *) argv[5];
cmask = (char *) argv[6];
badpix = (char *) argv[7];
broadalbtoa = (float *) argv[8];

/* Loop through each pixel in the arrays. If it is a "bad" pixel, skip it and assign the fluxes a
MISSING value. */

ERROR = FALSE;
for (i=0; i<nrows*ncols; i++) {
if (badpix[i] == (char) FALSE && solzen[i] <= DARKZEN) {
if (cmask[i] == (char) CLEAR)
broadalbtoa[i] = narr2bb_toa(ch1[i],ch2[i],solzen[i],surftype[i]);
}
else
broadalbtoa[i] = MISSING;
}

return ERROR;
}

/***************** CALL ANISOTROPIC REFLECTANCE CORRECTION *****************/

long call_anis_correct(short argc, void *argv[])

/*-------------------------------------------------------------------------
```

This function loops through the input arrays and calls 'anis_correct' (below) for each pixel. It is called from an IDL program with the following statement:

```
ERROR =
call_external(objfln,'call_anis_correct',ncols,nrows,broadalb,$ solzen,scan,relaz,surftype,cmask,badpix)
```

Input (as pointers to the specified type):

ncols, nrows : Dimensions of 'tsurf' and the arrays pointed to by 'array' [short integer]. While these arrays are 2D, they are treated here as 1D. If scalar values are passed in, both of these values must be 1. If a 1D array is passed in, then 'nrows' must be 1.

broadalb : Broadband albedo at TOA, in the range [0,1]. [float]

solzen : Solar zenith angle image (degrees) [float]

scan : Satellite scan angle (degrees) [float]

relaz : Satellite-sun relative azimuth angle (degrees), where 180 is looking into the sun and 0 is looking away (0..180) [float]

surftype : Surface type (global variables LAND, OCEAN, ICE, SNOW; see 'globals.pro' for values). LAND means snow-free land (vegetation cover), OCEAN means open water, ICE means sea ice (with or without snow), and SNOW means snow-covered land. However, corrections are currently done for only land and ice surface. Any other surface type uses the ice procedure. [char]

```
cmask : Cloud mask array, cloudy=CLOUDY, clear=CLEAR ('globals.h') [char]

badpix : Valid pixels array, where invalid=TRUE ('globals.h') [char]

Output:

broadalb : The broadband albedo corrected for anisotropy, in the range [0,1]. If badpix=TRUE
and/or the solar zenith angle is > DARKZEN, MISSING values are returned. For CLOUDY pixels
the input values are unchanged. Results outside of the range [0,1] are reset to 0 or 1. [fltarr or float]

Additional Variables (from globals.h):

TRUE, FALSE: true and false flags to be used for returning error condition

Called by: anis_correct (an IDL procedure)

Calls: anis_correct (the C function below)

----------------------------------------------------------------------------*/
{

/* parameters passed in from IDL */

short ncols, nrows;
char *cmask, *surftype, *badpix;
float *broadalb, *solzen, *scanang, *relaz;

/* local parameters */

long i;
long ERROR; /* Error flag, returned to IDL */
short j, k, l, m, index;

/* Check that correct number of parameters was passed */

if (argc != 9) {
fprintf(stdout, "argc = %d\r\n",argc);
fprintf(stdout, "ERROR >>> CALL_ANIS_CORRECT: INCORRECT NUMBER OF
ARGUMENTS\r\n");
ERROR = TRUE;
return ERROR;
}

/* Cast passed parameters to local vars. Copy the data table to fixed size
arrays. */
ncols = *(short *) argv[0];
nrows = *(short *) argv[1];
broadalb = (float *) argv[2];
solzen = (float *) argv[3];
scanang = (float *) argv[4];
relaz = (float *) argv[5];
surftype = (char *) argv[6];
cmask = (char *) argv[7];
badpix = (char *) argv[8];

/* Loop through each pixel in the arrays. If it is a "bad" pixel, skip it and assign the fluxes a
MISSING value. */

ERROR = FALSE;

for (i=0; i<nrows*ncols; i++) {
if (badpix[i] == (char) FALSE && solzen[i] <= DARKZEN) {
if (cmask[i] == (char) CLEAR)
broadalb[i] = anis_correct(broadalb[i],solzen[i],scanang[i],
relaz[i],surftype[i]);
}
else {
broadalb[i] = MISSING;
}
}

return ERROR;
}

/******CALL ANISOTROPIC REFLECTANCE CORRECTION WITH BLENDED OCEAN/ICE *******/

long call_anis_correct_blend(short argc, void *argv[])

/*--------------------------------------------------------------------------

This function loops through the input arrays and calls 'anis_correct_blend' (below) for each pixel. It
is called from an IDL program with the following statement:

ERROR = call_external(objfln,'call_anis_correct_blend',ncols,nrows,$ ch1, ch1_pure_ocean,
ch1_pure_ice,$broadalb,solzen,scan,relaz,surftype,cmask,badpix)

Input (as pointers to the specified type):
```

ncols, nrows : Dimensions of 'tsurf' and the arrays pointed to by 'array' [short integer]. While these arrays are 2D, they are treated here as 1D. If scalar values are passed in, both of these values must be 1. If a 1D array is passed in, then 'nrows' must be 1.

ch1 : Channels 1 albedos at TOA, in the range [0,1] and normalized for solar zenith angle and NOT corrected for anisotropy and the atmosphere. [float]

ch1_pure_ocean: The ch1 value which will be considered pure ocean. [scalar float]

ch1_pure_ice : The ch1 value which will be considered pure ocean. [scalar float]

broadalb : Broadband albedo at TOA, in the range [0,1]. [float]

solzen : Solar zenith angle image (degrees) [float]

scan : Satellite scan angle (degrees) [float]

relaz : Satellite-sun relative azimuth angle (degrees), where 180 is looking into the sun and 0 is looking away (0..180) [float]

surftype : Surface type (global variables LAND, OCEAN, ICE, SNOW; see 'globals.pro' for values). LAND means snow-free land (vegetation cover), OCEAN means open water, ICE means sea ice (with or without snow), and SNOW means snow-covered land. However, corrections are currently done for only land and ice surface. Any other surface type uses the ice procedure. [char]

cmask : Cloud mask array, cloudy=CLOUDY, clear=CLEAR ('globals.h') [char]

badpix : Valid pixels array, where invalid=TRUE ('globals.h') [char]

Output:

broadalb : The broadband albedo corrected for anisotropy with blending of albedos for OCEAN and ICE pixels, in the range [0,1]. If badpix=TRUE and/or the solar zenith angle is > DARKZEN, MISSING values are returned. For CLOUDY pixels the input values are unchanged. Results outside of the range [0,1] are reset to 0 or 1. [fltarr or float]

Additional Variables (from globals.h):

TRUE, FALSE : true and false flags to be used for returning error condition

Called by: anis_correct_blend (an IDL procedure)

Calls: anis_correct_blend (the C function below)

---*/

```
{

/* parameters passed in from IDL */

short ncols, nrows;
char *cmask, *surftype, *badpix;
float *ch1, ch1_pure_ocean, ch1_pure_ice;
float *broadalb, *solzen, *scanang, *relaz;

/* local parameters */

long i;
long ERROR; /* Error flag, returned to IDL */
short j, k, l, m, index;

/* Check that correct number of parameters was passed */

if (argc != 12) {
fprintf(stdout, "argc = %d\r\n",argc);
fprintf(stdout, "ERROR >>> CALL_ANIS_CORRECT_BLEND: INCORRECT NUMBER OF ARGUMENTS\r\n");
ERROR = TRUE;
return ERROR;
}

/* Cast passed parameters to local vars. Copy the data table to fixed size
arrays. */

ncols = *(short *) argv[0];
nrows = *(short *) argv[1];
ch1 = (float *) argv[2];
ch1_pure_ocean = *(float *) argv[3];
ch1_pure_ice = *(float *) argv[4];
broadalb = (float *) argv[5];
solzen = (float *) argv[6];
scanang = (float *) argv[7];
relaz = (float *) argv[8];
surftype = (char *) argv[9];
cmask = (char *) argv[10];
badpix = (char *) argv[11];
/* Loop through each pixel in the arrays. If it is a "bad" pixel, skip
it and assign the fluxes a MISSING value. */

ERROR = FALSE;

for (i=0; i<nrows*ncols; i++) {
if (badpix[i] == (char) FALSE && solzen[i] <= DARKZEN) {
if (cmask[i] == (char) CLEAR)
broadalb[i] = anis_correct_blend(ch1[i],ch1_pure_ocean,ch1_pure_ice,
broadalb[i],solzen[i],scanang[i],
relaz[i],surftype[i]);
}
else {
broadalb[i] = MISSING;
}
}

return ERROR;
}


/*********************** CALL ATMOSPHERIC CORRECTION *********************/

long call_atmos_correct(short argc, void *argv[])

/*---------------------------------------------------------------------------
```

This function loops through the input arrays and calls 'atmos_correct' (below) for each pixel. It is called from an IDL program with the following statement:

ERROR =
call_external(objfln,'call_atmos_correct',ncols,nrows,broadalb,$ solzen,wv,odhaze,surftype,cmask, badpix)

Input (as pointers to the specified type):

ncols, nrows : Dimensions of 'tsurf' and the arrays pointed to by 'array' [short integer]. While these arrays are 2D, they are treated here as 1D. If scalar values are passed in, both of these values must be 1. If a 1D array is passed in, then 'nrows' must be 1.

broadalb : Broadband TOA albedo, in the range [0,1]. [float]

solzen : Solar zenith angle image (degrees) [float] wv: Water vapor amount image (0.5 - 5.0 g/cm^2 = precipitable water in cm) [float]

odhaze : Aerosol optical depth (0.05 - 0.5) [float]

surftype : Surface type (global variables LAND, OCEAN, ICE, SNOW; see 'globals.pro' for values). LAND means snow-free land (vegetation cover), OCEAN means open water, ICE means sea ice (with or without snow), and SNOW means snow-covered land. However, corrections are currently done for only land and ice surface. Any other surface type uses the ice procedure. [char]

cmask : Cloud mask array, cloudy=CLOUDY, clear=CLEAR ('cmask.pro' and 'globals.h') [char]

badpix : Valid pixels array, where invalid=TRUE ('globals.h') [char]

Output:

broadalb : "Apparent" surface albedo corrected for the atmosphere in the range [0,1], which are directional, hemispherical (i.e., are not scan angle or azimuthally-dependent but do vary with the atmospheric conditions, hence "apparent"). If badpix=TRUE or the solar zenith angle > DARKZEN, MISSING values are returned. If the pixel is CLOUDY then the values that these variables have on input are unchanged. Results outside of the range [0,1] are reset to 0 or 1. [fltarr or float]

Additional Variables (from globals.h):

TRUE, FALSE : true and false flags to be used for returning error condition

Called by: atmos_correct (an IDL procedure)

Calls: atmos_correct (the C function below)

--*/

```
{

/* parameters passed in from IDL */

short ncols, nrows;
char *surftype, *cmask, *badpix;
float *solzen, *wv, odhaze;
float *broadalb;

/* local parameters */

long i;
long ERROR; /* Error flag, returned to IDL */
/* Check that correct number of parameters was passed */

if (argc != 9) {
fprintf(stdout, "argc = %d\r\n",argc);
fprintf(stdout, "ERROR >>> CALL_ATMOS_CORRECT: INCORRECT NUMBER OF
ARGUMENTS\r\n");
ERROR = TRUE;
return ERROR;
}

/* Cast passed parameters to local vars */

ncols = *(short *) argv[0];
nrows = *(short *) argv[1];
broadalb = (float *) argv[2];
solzen = (float *) argv[3];
wv = (float *) argv[4];
odhaze = *(float *) argv[5];
surftype = (char *) argv[6];
cmask = (char *) argv[7];
badpix = (char *) argv[8];

/* Loop through each pixel in the arrays. If it is a "bad" pixel, skip
it and assign the fluxes a MISSING value. */

ERROR = FALSE;

for (i=0; i<nrows*ncols; i++) {
if (badpix[i] == (char) FALSE && solzen[i] <= DARKZEN) {
if (cmask[i] == (char) CLEAR)
broadalb[i] = atmos_correct(broadalb[i],solzen[i],wv[i],odhaze,
surftype[i]);
}
else {
broadalb[i] = MISSING;
}
}

return ERROR;
}

/*********************** NARROW TO BROADBAND CONVERSION ********************/
```

float narr2bb_toa(float ch1, float ch2, float solzen, char surftype)

/*--

Converts the channels 1 and 2 reflectance/albedo values to a broadband albedo for clear conditions over snow/ice, vegetation (generic), and open water. This procedure operates on scalar values only.

Input:

ch1, ch2 : Channels 1 and 2 albedos at TOA, in the range [0,1], normalized for solar zenith angle, NOT corrected for the atmospheric attenuation, and corrected for anisotropic reflectance. [float]

solzen : The solar zenith angle (degrees) [float]

surftype : One byte (char) value representing the surface type (global variables LAND, OCEAN, ICE, SNOW* see 'globals.pro' for values). LAND means snow-free land (vegetation cover), OCEAN means open water, ICE means sea ice (with or without snow), and SNOW means snow-covered land. However, corrections are currently done for only land and ice surface. For other surface types the channel 1 reflectance is returned. [char]

Output as value of function:

Broadband TOA albedo in the range [0,1]. For surface types other than snow/ice or vegetation the clear sky albedo is returned. [fltarr or float] If badpix=TRUE and/or the solar zenith angle is > DARKZEN, MISSING values are returned. For CLOUDY pixels the input values are unchanged. Results outside of the range [0,1] are reset to 0 or 1.

Algorithm Summary:

The clear sky broadband surface albedo is computed as a function of the channels 1 and 2 reflectances. The regression relationship has the form:

broadalbtoa = a + b*rsurf1 + c*rsurf2

Two sets of coefficients were developed: one for snow-free land (generic vegetation) and one for snow/ice. Streamer was used to simulate AVHRR channels 1 and 2 albedos and a TOA broadband albedo for a variety of surface albedo, atmospheric, and viewing angle relationships. A linear regression procedures was then applied to the Streamer data. For open water we assume that the broadband albedo is the same as the channel 1 albedo (to a certain extent after Brieglieb et al. JCAM 1986, where the albedo is the same for all wavelengths).

--*/

```
{
float broadalbtoa;

/* If it's dark then the input reflectances were previously set to zero (in 'correct12'). Do the same with the broadband albedo.
Otherwise, compute it. */

if (ch1 == MISSING || ch2 == MISSING || solzen >= DARKZEN)
broadalbtoa = MISSING;

else if (surftype == (char) LAND)
broadalbtoa = 0.0404522 + 0.545025*ch1 + 0.299113*ch2;

else if ((surftype == (char) SNOW) || (surftype == (char) ICE))
broadalbtoa = 0.0215773 + 0.277479*ch1 + 0.506755*ch2;

else
broadalbtoa = ch1; /* Open water */

if ((broadalbtoa < 0.0) && (broadalbtoa != MISSING)) broadalbtoa = 0.0;

return broadalbtoa;
}


/************************ ANISOTROPIC CORRECTION *****************************/

float anis_correct(float broadalb, float solzen, float scan, float relaz, char surftype)

/*------------------------------------------------------------------------------

Corrects the broadband TOA albedo data for anisotropic reflectance. This procedure operates on scalar values only.

Input:

broadalb : TOA broadband albedo, in the range [0,1]. [float]

solzen : Solar zenith angle image (degrees) [float]

scan : Satellite scan angle (degrees) [float]

relaz : Satellite-sun relative azimuth angle (degrees), where 180 is looking into the sun and 0 is looking away (0..180) [float]
```

surftype : One byte (char) value representing the surface type (global variables LAND, OCEAN, ICE, SNOW see 'globals.h' for values). LAND means snow-free land (vegetation cover), OCEAN means open water, ICE means sea ice (with or without snow), and SNOW means snow-covered land. However, corrections are currently done for only snow-free land, ocean, and snow/ice (the same) surfaces. Any other surface type uses the snow procedure.

Static data in anisdata.h:

rmatrx : Anisotropic reflectance values (not quite factors) based on ERBE and GOES, dimension: (3,10,7,8) for eight relative azimuth bins, seven viewing zenith bins, ten solar zenith bins, and three scene types. [float]

albmn : Normalizing factors for 'albmn', dimension: (3,10), for ten solar zenith bins and three scene types. [float]

Output as value of function:

The broadband albedo corrected for anisotropy, in the range [0,1]. If badpix=TRUE and/or the solar zenith angle is > DARKZEN, MISSING values are returned. For CLOUDY pixels the input values are unchanged. Results outside of the range [0,1] are reset to 0 or 1. [fltarr or float]

Algorithm Summary:

The anisotropic reflectance factors are based on ERBE and GOES data for clear ice (clear sky over ice), clear land, and clear ocean as given in NASA Reference Publication 1184, July 1988, "Angular Radiation Models for Earth-Atmosphere System, Volume I-Shortwave Radiation", by Suttles et al. The anisotropic reflectance factors are obtained via interpolation in the table, and are divided into the TOA reflectance to produce a TOA albedo. Data were provided by L. Smith (NASA/Langley).

```
-------------------------------------------------------------------------*/
{
short ISCENE, ITEMP, IVZEN, ISZEN, IRAZ;
float VZEN, CSUN, RAZ, FVZEN, CVZEN, FSZEN, CSZEN, FRAZ, CRAZ, TEMP, F, bbalb;

float rearth = 6357.0; /* Radius of the earth */
float rehsat = 7207.0; /* Rearth + Hsat(850 km) */

short MVZEN[] = {0,0,0,0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5};
float GVZEN[] = {0.,21.,33.,45.,57.,69.,90};
short MSZEN[] = {0,0,0,1,1,2,2,3,3,4,4,5,5,6,6,7,7,8,8,8};
float GSZEN[] = {0.,.15,.25,.35,.45,.55,.65,.75,.85,1.00};
```

```
short MRAZ[] = {0,0,0,0,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,6,6,6,6};
float GRAZ[] = {0.,20.,45.,75.,105.,135.,160.,180.};

/* If it's dark or if the input value is missing, return MISSING. */

if (solzen >= DARKZEN || broadalb == MISSING) return MISSING;

/* Convert values to what the interpolation code expects: VZEN is viewing
zenith angle (0-90 deg), CSUN is the solar zenith angle cosine (0-1),
RAZ is the relative azimuth angle (0-180 deg, 0 is looking into the sun
as in Streamer, but opposite of CASPR input), ISCENE is the scene type
(0=clear ocean, 1=clear land, 2=clear snow/ice). */

if (surftype == (char) OCEAN)
ISCENE = 0;
else if (surftype == (char) LAND)
ISCENE = 1;
else
ISCENE = 2;
RAZ = 180.0 - relaz;
VZEN = asin((sin(scan*DEGRAD)*rehsat)/rearth)*RADDEG;
CSUN = cos(solzen*DEGRAD);

/* Perform tri-linear interpolation in the table. First determine the indices
and fractions. Then linearly interpolate. */


ITEMP = VZEN/3.000001;
IVZEN = MVZEN[ITEMP];
FVZEN = (VZEN - GVZEN[IVZEN]) / (GVZEN[IVZEN+1] - GVZEN[IVZEN]);
CVZEN = 1. - FVZEN;

TEMP = 1. - CSUN;
ITEMP = TEMP/.05000001;
ISZEN = MSZEN[ITEMP];
FSZEN = (TEMP - GSZEN[ISZEN]) / (GSZEN[ISZEN+1] - GSZEN[ISZEN]);
CSZEN = 1. - FSZEN;


ITEMP = RAZ/5.000001;
IRAZ = MRAZ[ITEMP];
FRAZ = (RAZ - GRAZ[IRAZ]) / (GRAZ[IRAZ+1] - GRAZ[IRAZ]);
CRAZ = 1. - FRAZ;
```

```
F = (CVZEN*CRAZ*rmatrx[ISCENE][ISZEN][IVZEN][IRAZ]
+ FVZEN*CRAZ*rmatrx[ISCENE][ISZEN][IVZEN+1][IRAZ]
+ CVZEN*FRAZ*rmatrx[ISCENE][ISZEN][IVZEN][IRAZ+1]
+ FVZEN*FRAZ*rmatrx[ISCENE][ISZEN][IVZEN+1][IRAZ+1] )
* CSZEN / albmn[ISCENE][ISZEN]
+(CVZEN*CRAZ*rmatrx[ISCENE][ISZEN+1][IVZEN][IRAZ]
+ FVZEN*CRAZ*rmatrx[ISCENE][ISZEN+1][IVZEN+1][IRAZ]
+ CVZEN*FRAZ*rmatrx[ISCENE][ISZEN+1][IVZEN][IRAZ+1]
+ FVZEN*FRAZ*rmatrx[ISCENE][ISZEN+1][IVZEN+1][IRAZ+1] )
* FSZEN / albmn[ISCENE][ISZEN+1];

/* Convert the albedo. */

bbalb = broadalb / F;


if (bbalb < 0.) bbalb = 0.;

return bbalb;
}

/********** ANISOTROPIC CORRECTION WITH OCEAN/ICE BLENDING *****************/

float anis_correct_blend(float ch1, float ch1_pure_ocean, float ch1_pure_ice,
float broadalb, float solzen, float scan,
float relaz, char surftype)

/*-------------------------------------------------------------------------

Corrects the broadband TOA albedo data for anisotropic reflectance using a weighted sum blend of
ocean and ice albedos for ocean and ice surface types. This procedure operates on scalar values
only.

Input:

ch1: Channel 1 albedo at TOA , in the range [0,1], normalized for solar zenith angle, NOT corrected
for the atmospheric attenuation, and corrected for anisotropic reflectance. [float]

ch1_pure_ocean: The ch1 value which will be considered pure ocean. [float]

ch1_pure_ice : The ch1 value which will be considered pure ocean. [float]


broadalb,
solzen,
scan,
relaz,
surftype : See anis_correct.

Algorithm Summary:
If surface type is LAND, or ch1 <= ch1_pure_ocean,
or ch1 >= ch1_pure_ice, anis_correct is called
with the given broadalb, solzen, scan, relaz, and surftype
parmameters and the return value from anis_correct is
returned without modification. Otherwise, anis_correct is
called twice with the given broadalb, solzen, scan, and relaz
parameters, but with the surftype parameter forced to OCEAN
on the first call and ICE on the second call. A weighted
sum average of the returned OCEAN and ICE albedos is then
calculated and returned.

The weighted sum is computed as follows:

range = ch1_pure_ice - ch1_pure_ocean
ocean_weight = (ch1_pure_ice - ch1) / range
ice_weight = 1.0 - ocean_weight
blended_albedo = ocean_albedo * ocean_weight +
ice_albedo * ice_weight
----------------------------------------------------------------------*/
{
float range;
float ocean_weight;
float ice_weight;
float ocean_albedo;
float ice_albedo;
float blended_albedo;

/* If it's dark or if the input value is missing, return MISSING. */

if (solzen >= DARKZEN || broadalb == MISSING) return MISSING;

if (surftype == (char)LAND || surftype == (char)SNOW ||
ch1 <= ch1_pure_ocean || ch1 >= ch1_pure_ice)
return(anis_correct(broadalb, solzen, scan, relaz, surftype));
range = ch1_pure_ice - ch1_pure_ocean;
ocean_weight = (ch1_pure_ice - ch1) / range;
ice_weight = 1.0 - ocean_weight;

ocean_albedo = anis_correct(broadalb, solzen, scan, relaz, (char)OCEAN);
ice_albedo = anis_correct(broadalb, solzen, scan, relaz, (char)ICE);
blended_albedo = ocean_albedo * ocean_weight + ice_albedo * ice_weight;

if (blended_albedo < 0.0)
blended_albedo = 0.0;

return blended_albedo;
}

/*************************** ATMOSPHERIC CORRECTION ***********************/


float atmos_correct(float broadalb, float solzen, float wv, float odhaze,
char surftype)
/*----------------------------------------------------------------
Converts a TOA broadband albedo to a surface broadband albedo. The
apparent (varies with atmospheric conditions) albedo will be calculated.
The apparent albedo is what would be measured with a pair of up/down
radiometers given arbitrary atmospheric conditions. The inherent albedo,
which is not calculated here (but could be), is an albedo without an
atmosphere; i.e., it doesn't depend upon gases and aerosols.

This procedure operates on scalar values only.


Input:
broadalb : TOA broadband albedo (corrected for anisotropy), in the range
[0,1].
solzen : Solar zenith angle, in degrees.
wv: Water vapor amount (0.5 - 5.0 g/cm^2
= precipitable water in cm)
odhaze: Aerosol optical depth (0.05 - 0.5)
surftype : One byte (char) value representing the surface type (global
variables LAND, OCEAN, ICE, SNOW see 'globals.h' for
values). LAND means snow-free land (vegetation cover),
OCEAN means open water, ICE means sea ice (with or without
snow), and SNOW means snow-covered land.
However, corrections are currently done for only snow-free
```

land, ocean, and snow/ice (the same) surfaces. Any other surface type uses the snow procedure.

Output as value of function:
"Apparent" surface albedo corrected for the atmosphere in the range [0,1], which are directional, hemispherical (i.e., are not scan angle or azimuthally-dependent but do vary with the atmospheric conditions, hence "apparent"). If badpix=TRUE or the solar zenith angle > DARKZEN, MISSING values are returned. If the pixel is CLOUDY then the values that these variables have on input are unchanged. Results outside of the range [0,1] are reset to 0 or 1. [fltarr or float]

Algorithm Summary:

The correction for atmospheric attenuation in in the broadband albedo was developed with data generated by Streamer for two surface types (snow and generic vegetation combined) and a variety of atmospheric conditions (global). It is a function of solar zenith angle, water vapor, and aerosol optical depth. Aerosol optical depth and water vapor amounts are accounted for explicitly through linear interpolation of coefficients (which probably introduces some error); ozone amount is prescribed (0.24 cm). The form of the correction is:

surf_alb = (toa_alb - a) / b

where surf_alb is the atmospherically corrected broadband surface albedo, toa_alb is the TOA broadband albedo, and a and b are determined through regression using the Streamer data.

Unfrozen ocean uses a simpler approach, where the surface albedo is related to the TOA albedo, zenith angle, aerosol optical depth, and precipitable water by

surf_alb = a + b*toa_alb+ c*cos(zen) + d*pw + e*aertau

where a, b, c, d, and e are determined empirically using modeled albedos. Ocean surface albedos less than 0.04 are set to that value.

The retrieved surface albedo is an "apparent" albedo; i.e., that which would be measured with radiometers and which vary with changes in

```
atmospheric conditions, especially for bright surfaces.
-----------------------------------------------------------------------*/
{
int nsols = 20;
float bbalb;
float sqrtpw, broadalb2, alpha1, alpha2, beta1, beta2, alpha, beta;

/* These coefficients are for 'nsols' solar zenith angle cosines from 1 down
to 0.05 in increments of 0.05. The first dimension corresponds to the
solar zenith angles in the order shown below. The second dimension holds
the 'a' and 'b' coefficients, in that order.


All calculations are for an ozone amount of 325 Dobson units.

The solar Zenith angles are as follows.
Cosines:
1.00, 0.95, 0.90, 0.85, 0.80, 0.75, 0.70, 0.65, 0.60, 0.55,
0.50, 0.45, 0.40, 0.35, 0.30, 0.25, 0.20, 0.15, 0.10, 0.05
Degrees:
0.00, 18.19, 25.84, 31.79, 36.87, 41.41, 45.57, 49.46, 53.13, 56.63,
60.00, 63.26, 66.42, 69.51, 72.54, 75.52, 78.46, 81.37, 84.26, 87.13 */

float odmin,odmax;
float wvmin,wvmax;
float odrange,oddiff,odscale;
float wvrange,wvdiff,wvscale;
int k;
float a1,b1,a2,b2,awv1,bwv1,awv2,bwv2;
float a,b;
float coszen;
int zen_index;

/*------ COEFFICIENTS FOR APPARENT ALBEDO OF SNOW/ICE AND VEGETATION -------*/

/* First for a water vapor amount of 0.5 g/cm^2 and aerosol optical
depth 0.05 (aer1_wv1), then wv=0.5, aer=0.5 (aer2_wv1), then
wv=5.0, aer=0.05 (aer1_wv2), then wv=5.0, aer=0.5 (aer2_wv2). */

float aer1_wv1[20][2] =
{ { 0.0369468, 0.7716573}, { 0.0393092, 0.7681513},
{ 0.0420182, 0.7638956}, { 0.0449436, 0.7596443},
{ 0.0480998, 0.7549872}, { 0.0515801, 0.7498252},
{ 0.0556449, 0.7435693}, { 0.0598915, 0.7376858},
{ 0.0648356, 0.7302867}, { 0.0702911, 0.7221886},
{ 0.0770464, 0.7122877}, { 0.0843299, 0.7015929},
{ 0.0933098, 0.6879621}, { 0.1038388, 0.6723217},
{ 0.1166860, 0.6532161}, { 0.1327160, 0.6291631},
{ 0.1532862, 0.5983471}, { 0.1807689, 0.5560154},
{ 0.2196831, 0.4951470}, { 0.2790609, 0.3983846} };

float aer2_wv1[20][2] =
{ { 0.0440627, 0.4562957}, { 0.0491487, 0.4497921},
{ 0.0544951, 0.4427959}, { 0.0607027, 0.4344482},
{ 0.0671587, 0.4260309}, { 0.0741401, 0.4170585},
{ 0.0819679, 0.4068565}, { 0.0903022, 0.3963189},
{ 0.0995805, 0.3844509}, { 0.1100535, 0.3704767},
{ 0.1216115, 0.3557419}, { 0.1344762, 0.3394325},
{ 0.1491109, 0.3203476}, { 0.1654393, 0.2998796},
{ 0.1841938, 0.2767967}, { 0.2054262, 0.2506457},
{ 0.2296357, 0.2223978}, { 0.2571991, 0.1917805},
{ 0.2878296, 0.1623356}, { 0.3209661, 0.1382332} };

float aer1_wv2[20][2] =
{ { 0.0301825, 0.7455254}, { 0.0328166, 0.7420131},
{ 0.0358166, 0.7377133}, { 0.0390927, 0.7331624},
{ 0.0426720, 0.7282085}, { 0.0464936, 0.7231386},
{ 0.0507808, 0.7172813}, { 0.0556268, 0.7107663},
{ 0.0611204, 0.7033693}, { 0.0675476, 0.6945344},
{ 0.0745468, 0.6851625}, { 0.0830090, 0.6737533},
{ 0.0926029, 0.6612120}, { 0.1043418, 0.6455948},
{ 0.1183378, 0.6274127}, { 0.1360318, 0.6041222},
{ 0.1584471, 0.5754020}, { 0.1884491, 0.5367222},
{ 0.2302195, 0.4839584}, { 0.2928393, 0.4020659} };

float aer2_wv2[20][2]=
{ { 0.0146276, 0.6222795}, { 0.0217345, 0.6142944},
{ 0.0296301, 0.6050117}, { 0.0379796, 0.5953020},
{ 0.0471900, 0.5848151}, { 0.0569098, 0.5736121},
{ 0.0678248, 0.5610381}, { 0.0795343, 0.5476506},
{ 0.0924229, 0.5327814}, { 0.1066507, 0.5160899},
{ 0.1222040, 0.4982739}, { 0.1395290, 0.4782666},
{ 0.1589199, 0.4553649}, { 0.1801602, 0.4307515},
{ 0.2038677, 0.4030677}, { 0.2301649, 0.3719538},
{ 0.2588158, 0.3378740}, { 0.2892141, 0.3007274},
{ 0.3194277, 0.2621159}, { 0.3452438, 0.2208999} };

/*------------------------------------------------------------------------*/


/* If it's dark or if the input value is missing or if water vapor is
missing, return MISSING. */

if (solzen >= DARKZEN || broadalb == MISSING || wv == MISSING) return MISSING;

coszen = cos(solzen*DEGRAD);
if (coszen < 0.0501) coszen = .0501; /* Limit of solar angles */

/* SNOW, ICE, or LAND */

if (surftype != (char) OCEAN) {

zen_index = (int)(((1.0 - coszen)/0.05)+0.5);

/* The min and max aerosol optical depth and water vapor amounts
for the coefficients specified above. */

odmin=0.05;
odmax=0.5;
wvmin=0.5;
wvmax=5.0;


/* Determine the proper scale factor for the specified aerosol optical
depth and the water vapor amount. */


odrange = odmax - odmin;
oddiff = odhaze - odmin;
odscale = oddiff/odrange;
wvrange = wvmax - wvmin;
wvdiff = wv - wvmin;
wvscale = wvdiff/wvrange;

/* Get the estimated coefficients for the specified haze optical depth by
linearly interpolating between those for the optical depths in the
data statements above. This is done for each of the two water
vapor amounts. The final coefficients are then obtained by
```

```
interpolating (again linearly) between those computed for the desired
optical depth at the two water vapor extremes. */

a1 = aer1_wv1[zen_index][0];
b1 = aer1_wv1[zen_index][1];
a2 = aer2_wv1[zen_index][0];
b2 = aer2_wv1[zen_index][1];
awv1 = a1 + odscale*(a2 - a1);
bwv1 = b1 + odscale*(b2 - b1);


a1 = aer1_wv2[zen_index][0];
b1 = aer1_wv2[zen_index][1];
a2 = aer2_wv2[zen_index][0];
b2 = aer2_wv2[zen_index][1];
awv2 = a1 + odscale*(a2 - a1);
bwv2 = b1 + odscale*(b2 - b1);


a = awv1 + wvscale*(awv2 - awv1);
b = bwv1 + wvscale*(bwv2 - bwv1);

bbalb = (broadalb - a) / b;
}

/* OCEAN */

else {
bbalb = -0.112236 + broadalb*0.948389 + coszen*0.108496 + wv*0.00242575 -
odhaze*0.125026;
}

/*---------- ALTERNATE METHOD BY LI & GARAND, ALL SURFACE TYPES ------------*/
/*---------- (hasn't been tested in this function) Uncomment to use.

coszen = cos(solzen*DEGRAD);
sqrtpw = sqrt(wv)
broadalb2 = broadalb * 100.

alpha1 = -0.96882 + 0.718*sqrtpw;
alpha2 = -4.11460 - 0.76347*sqrtpw;
beta1 = 1.16711 + 0.05963*sqrtpw;
beta2 = 0.07514 + 0.04105*sqrtpw;
alpha = alpha1 + alpha2/coszen;
```

```
beta = beta1 + beta2/coszen;
bbalb = alpha + beta*broadalb2;
bbalb = bbalb/100.0;
--------------------------------------------------------------------*/

if (bbalb < 0.04) bbalb = 0.04;

return bbalb;
}
```

APPENDIX C: RETRIEVAL OF PRECIPITABLE WATER SOURCE CODE DOCUMENT

Procedures in this file are for the retrieval of precipitable water from AVHRR. They are:

call_precip_water

precip_water

Compilation: See the file README.C for instructions on compiling this file.

```
#include <assert.h>

#include <errno.h>

#include <stdio.h>

#include <math.h>

#include "globals.h"

long call_precip_water_avhrr(short argc, void *argv[]);

float precip_water_avhrr(float tb4, float tb5, float scan);


/************************* CALL PRECIPITABLE WATER *************************/


long call_precip_water_avhrr(short argc, void *argv[])


/*-------------------------------------------------------------------------

This function loops through the input arrays and calls 'precip_water' (below) for each pixel. It is
called from an IDL program with the following statement:


ERROR = call_external(objfln,'call_precip_water_avhrr',ncols,nrows,tb4,tb5,$
scanang,cmask,badpix,pw)


Input (as pointers to the specified type):


ncols, nrows : Dimensions of 'tsurf' and the arrays pointed to by 'array' [short integer]. While these
arrays are 2D, they are treated here as 1D. If scalar values are passed in, both of these values
must be 1. If a 1D array is passed in, then 'nrows' must be 1.
```

```
tb4,tb5: AVHRR channels 4 and 5 brightness temperatures (K) [float]

scanang: Sensor scan angle (degrees) [float]

cmask: Cloud mask array, cloudy=CLOUDY, clear=CLEAR ('globals.h') [char]

badpix: Valid pixels array, where invalid=TRUE ('globals.h') [char]

Output:

pw: Total precipitable water (cm) [float] If badpix=TRUE, MISSING values are returned. If the pixel
is CLOUDY then the values that this variable has on input is unchanged.

Additional Variables (from globals.h):

TRUE, FALSE : true and false flags to be used for returning error condition

Called by: precip_water_avhrr (an IDL procedure)

Calls: precip_water_avhrr (the C function below)

-------------------------------------------------------------------------*/
{
/* parameters passed in from IDL */

short ncols, nrows;
char *cmask, *badpix;
float *tb4, *tb5, *scanang, *pw;

/* local parameters */

long i;
long ERROR; /* Error flag, returned to IDL */

/* Check that correct number of parameters was passed */

if (argc != 8) {
fprintf(stdout, "argc = %d\r\n",argc);
fprintf(stdout, "ERROR >>> CALL_PRECIP_WATER: INCORRECT NUMBER OF ARGUMENTS\r\n");
ERROR = TRUE;
return ERROR;
}

/* Cast passed parameters to local vars */

ncols = *(short *) argv[0];
nrows = *(short *) argv[1];
tb4 = (float *) argv[2];
tb5 = (float *) argv[3];
scanang = (float *) argv[4];
cmask = (char *) argv[5];
badpix = (char *) argv[6];
pw = (float *) argv[7];

/* Loop through each pixel in the arrays. If it is a "bad" pixel, skip it and assign the fluxes a MISSING value.*/

ERROR = FALSE;

for (i=0; i<nrows*ncols; i++) {
if (badpix[i] != (char) TRUE) {
pw[i] = precip_water_avhrr(tb4[i],tb5[i],scanang[i]);
if ( pw[i] > 5. ) pw[i] = 5. ;
if ( pw[i] < .5 ) pw[i] = .5 ;
if ( pw[i] == MISSING ) badpix[i] = 1 ;

}

else

pw[i] = MISSING;

}


return ERROR;

}

/*************************** PRECIPITABLE WATER **************************/


float precip_water_avhrr(float tb4, float tb5, float scan)


/*------------------------------------------------------------------------

Estimates the clear-sky total precipitable water from the split-window channels, based on a
regression of Lowtran-modelled brightness temps and precipitable water from rawinsonde data.
This procedure operates on scalar values only.


Input:


tb4,tb5 : Channels 4 and 5 brightness temperatures (K)


scan : Sensor scan angle (degrees)


Output:


pw: Total precipitable water (cm)


Called by: call_precip_water


Calls: None


Algorithm Summary:
```

The total precipitable water (cm) in the atmospheric vertical column is estimated using the split-window thermal channels. The relationship is:

PW = exp[b0 + b1(T4-T5) + b2(T5)] cos(theta)

where Ti is the brightness temperature of an AVHRR channel i and theta is the sensor scan angle. The coefficients b0, b1, and b2 were determined over a range of surface temperatures and water vapor amounts using AVHRR radiances modelled with LOWTRAN-7. Arctic rawinsonde data were employed. This procedure is for clear-sky pixels only. Initially converted to C from IDL by Todd Hutchinson, University of Colorado. Further modified by J. Key

```
----------------------------------------------------------------------------*/
{
float pw;
float tmp;
/* if the difference between tb4 and tb5 is too large, it can cause pw to be
too large and the program may core dump. Too eliminate this, set pw to
missing if tb4-tb5 too large. In the calculation of pw, the
division by 10 is necessary to convert mm to cm. */

tmp=tb4-tb5;
if ((tmp < -3.)||(tmp > 10.)) pw = MISSING;
else {
tmp= -10.4974 + 0.751008*tmp + 0.0453005*tb5;
pw = exp(tmp) * cos(scan*DEGRAD) / 10.;
}

return pw;
}
```

APPENDIX D: RETRIEVAL OF SURFACE TEMPERATURE

```
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++

 Procedures in this file are for the retrieval of surface temperature.

 They are:

   call_surftemp_clear

   surftemp_clear


 Compilation:  See the file README.C for instructions on compiling this file.

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++*/


#include <assert.h>

#include <errno.h>

#include <stdio.h>

#include <math.h>

#include "globals.h"


long call_surftemp_clear(short argc, void *argv[]);

float surftemp_clear (float tb4, float tb5, float scanang, short hemis,

                      short satellite, char surftype);



/******************** CALL CLEAR SKY SURFACE TEMPERATURE ********************/


long call_surftemp_clear(short argc, void *argv[])

/*------------------------------------------------------------------------------

  This function loops through the input arrays and calls 'surftemp_clear'

  (below) for each pixel.  It is called from an IDL program with the

  following statement:


  ERROR = call_external(objfln,'call_surftemp_clear',ncols,nrows,tb4,tb5,$
                 scanang,hemis,satellite,surftype,cmask,badpix,tsurf)

Input (as pointers to the specified type):
  ncols, nrows   : Dimensions of 'tsurf' and the arrays pointed to by 'array'
                 [short integer].  While these arrays are 2D, they are
                 treated here as 1D.  If scalar values are passed in,
                 both of these values must be 1.  If a 1D array is passed
                 in, then 'nrows' must be 1.
  tb4,tb5        : AVHRR channels 4 and 5 brightness temperatures (K) [float]
  scanang        : Sensor scan angle (degrees) [float]
  hemis          : Hemisphere (0=southern,1=northern) [short integer]
  satellite      : satellite number (7,9,11,12,14,15(K),16(L))  [short integer]
  surftype       : Surface type as LAND, OCEAN, ICE, SNOW (see 'globals.pro' for
                 values).  LAND means snow-free land (vegetation cover),
                 OCEAN means open water, ICE means sea ice (with or without
                 snow), and SNOW means snow-covered land.
                 However, corrections are currently done for only land
                 and ice surfaces.  Any other surface type uses the ice
                 procedure. [char]
  cmask          : Cloud mask array, cloudy=CLOUDY, clear=CLEAR ('globals.h')
                 [char]
  badpix         : Valid pixels array, where invalid=TRUE ('globals.h') [char]

Output:
  tsurf          : Surface temperature (K) [float]
                 If badpix=TRUE, MISSING values are returned.  If the
                 pixel is CLOUDY then the values that this variable
                 has on input is unchanged.

Additional Variables (from globals.h):
  TRUE, FALSE    : true and false flags to be used for returning error
                condition
  Called by: surftemp_clear (an IDL procedure)

  Calls: surftemp_clear (the C function below)
--------------------------------------------------------------------------*/
{

/*  parameters passed in from IDL  */

  short  ncols, nrows, hemis, satellite;
  char   *cmask, *surftype, *badpix;
  float  *tb4, *tb5, *scanang, *tsurf;

/*  local parameters  */

  long i;
  long  ERROR;                 /* Error flag, returned to IDL */

/*  Check that correct number of parameters was passed  */

  if (argc != 11) {
    fprintf(stdout, "argc = %d\r\n",argc);
    fprintf(stdout, "ERROR >>> CALL_SURFTEMP_CLEAR: INCORRECT NUMBER OF
ARGUMENTS\r\n");
    ERROR = TRUE;
    return ERROR;
  }

/*  Cast passed parameters to local vars  */

  ncols     = *(short *) argv[0];
  nrows     = *(short *) argv[1];
  tb4       = (float *) argv[2];
  tb5       = (float *) argv[3];
  scanang   = (float *) argv[4];
  hemis     = *(short *) argv[5];
  satellite = *(short *) argv[6];
  surftype  = (char *) argv[7];
  cmask     = (char *) argv[8];
  badpix    = (char *) argv[9];
  tsurf     = (float *) argv[10];

/*  Loop through each pixel in the arrays.  If it is a "bad" pixel, skip
    it and assign the fluxes a MISSING value.  */

  ERROR = FALSE;

  for (i=0; i<nrows*ncols; i++) {
    if (badpix[i] != (char) TRUE) {
      if (cmask[i] == (char) CLEAR)
        tsurf[i] = surftemp_clear(tb4[i],tb5[i],scanang[i],hemis,satellite,
                                  surftype[i]);
    }
    else
      tsurf[i] = MISSING;
  }

return ERROR;
}



/********************** CLEAR SKY SURFACE TEMPERATURE ***********************/

float surftemp_clear (float tb4, float tb5, float scanang, short hemis,
                      short satellite, char surftype)
```

```
/*------------------------------------------------------------------------
 Surface temperature retrieval for clear sky conditions.

 This procedure operates on scalar values only.

 Input:
   tb4       - Channel 4 brightness temperature (K)
   tb5       - Channel 5 brightness temperature (K)
   scanang   - scan angle (degrees)
   hemis     - hemisphere (0=southern,1=northern)
   satellite - satellite number (7,9,11,12,14,15(K),16(L))
   surftype  - A value representing the surface type (global variables
             LAND, OCEAN, ICE, SNOW* see 'globals.h' for values).
             LAND means snow-free land (vegetation cover), OCEAN means
             open water, ICE means sea ice (with or without snow), and
             SNOW means snow-covered land.
             However, corrections are currently done for only land
             and ice surface.  Any other surface type uses the ice
             procedure.

 Returned as value of function:
   tsurf     - Clear sky surface temperature (K)

 Calls: None

 Called by: call_surftemp_clear

 Algorithm Summary:

 The clear sky temperature of ice and snow (and currently open ocean)
 is estimated as a function of the split-window brightness temperature
 difference, the channel 4 temperature, and the sensor scan angle.
```

```
  The algorithm uses coefficients for three temperature ranges and is
  based on simulations using LOWTRAN with Arctic Ocean and Antarctic
  radiosonde data.  The form of the correction equation is:

      tsurf = a + b*Tb4 + c*(Tb4-Tb5) + d*(Tb4-Tb5)*(sec(scanang)-1)

  where there is a unique set of coefficients a, b, c, and d for the
  temperature ranges: < 240K, 240-260K, and > 260K.

  For snow-free land surfaces a somewhat different equation is used:

      tsurf = a + b*Tb4 + c*Tb5 + e*emiss4 + f*emiss5

  where 'emiss4' and 'emiss5' are the emissivities of the surface in
  channels 4 and 5.  The coefficients in this equation are also derived
  from a regression analysis of modelled radiances, but radiosonde profiles
  from only Arctic land areas are used in the analysis.  The two emissivities
  are set to constant values below.  If other emissivities are desired,
  this file must be edited.
-------------------------------------------------------------------------*/
{
  short sat;
  short temprange;
  float tsurf;

/* SNOW-ICE COEFFICIENTS:
   a,b,c,d (i,j,k)  where i=0,1 hemisphere: 0=Antarctic,1=Arctic
                          j=0..2 temp range: 0 = <240, 1 = 240-260, 2 = >260
                          k=0..6 NOAA satellite: 7,9,11,12,14,15,16  */

  float a[2][3][7] =
  {{{-1.216191, -1.762823, -1.466105, -0.800189, -1.180768, -1.258905, -0.292169},
    {-6.400722, -8.083512, -7.100426, -4.823715, -5.647473, -6.381229, -3.650344},
    {-7.000356, -7.985411, -7.398464, -6.114496, -6.602028, -7.036723, -5.768286}},

   {{-3.824689, -5.482065, -4.655325, -2.798275, -3.846001, -4.034918, -1.453829},
    {-4.605042, -6.541145, -5.393341, -3.475960, -4.397758, -4.728985, -2.304536},
    {-4.415813, -5.254914, -4.769341, -4.121089, -4.498451, -4.633262, -3.676576}}};

  float b[2][3][7] =
  {{{1.004336, 1.007446, 1.005666, 1.002277, 1.003951, 1.004476, 1.000107},
    {1.025619, 1.032868, 1.028625, 1.019081, 1.022697, 1.025568, 1.013983},
    {1.027369, 1.031757, 1.029138, 1.023614, 1.025938, 1.027533, 1.021676}},

   {{1.014527, 1.021786, 1.018096, 1.010394, 1.014939, 1.015444, 1.004601},
    {1.017617, 1.025859, 1.020959, 1.013116, 1.017179, 1.018188, 1.008018},
    {1.016484, 1.020431, 1.018134, 1.015023, 1.017059, 1.017383, 1.012527}}};

  float c[2][3][7] =
  {{{1.365568, 0.477679, 1.092875, 1.729552, 1.616824, 1.430976, 1.506015},
    {0.981037, 0.600573, 0.857094, 1.138661, 0.919952, 0.973153, 1.181797},
    {1.079770, 0.921390, 1.035730, 1.174924, 0.963384, 1.085787, 1.215382}},

   {{2.228758, 1.995829, 2.196793, 2.100035, 1.904878, 2.222340, 1.946308},
    {1.795312, 1.647280, 1.763992, 1.681569, 1.411066, 1.757905, 1.637119},
    {1.666478, 1.635749, 1.664891, 1.669003, 1.402800, 1.659278, 1.690164}}};

  float d[2][3][7] =
  {{{-0.650609, -0.080109, -0.477556, -0.757764, -0.729174, -0.661893, -0.566436},
    { 0.562568,  1.158426,  0.766613,  0.383119,  0.552105,  0.607435, 0.194899},
    { 0.889363,  1.433513,  1.073912,  0.676141,  0.793529,  0.920915, 0.477659}},

   {{-1.294083, -1.183647, -1.268939, -1.027160, -0.879747, -1.228140, -0.907209},
    {-0.080299,  0.278680,  0.041162, -0.018824,  0.213224, -0.001810, -0.131031},
    { 0.684021,  1.147774,  0.847497,  0.547265,  0.690478,  0.733954, 0.347890}}};

/* SNOW-FREE LAND COEFFICIENTS:
   al,bl,cl,dl,el (j,k) where
         j=0..2 temp range: 0 = <240, 1 = 240-260, 2 = >260
         k=0..6 NOAA satellite: 7,9,11,12,14,15,16  */

  float al[3][7] =
   {{26.0309, 23.0055, 24.5757, 29.1836, 27.8690, 26.0307, 31.7447},
    {32.1194, 29.3755, 30.9222, 34.8680, 33.5189, 32.0778, 37.6640},
    {44.4224, 41.5469, 43.0879, 46.9049, 45.9367, 44.1398, 49.0214}};

  float bl[3][7] =
   {{4.0147, 4.4368, 4.2369, 3.4836, 3.2520, 3.9893, 3.2120},
    {3.5683, 3.6499, 3.5992, 3.5896, 3.2796, 3.5843, 3.5496},
    {3.6507, 3.7915, 3.7034, 3.6529, 3.3803, 3.6687, 3.5950}};

  float cl[3][7] =
   {{-2.9919, -3.4103, -3.2127, -2.4606, -2.2260, -2.9655, -2.1935},
    {-2.5444, -2.6167, -2.5714, -2.5732, -2.2587, -2.5600, -2.5422},
    {-2.6387, -2.7710, -2.6874, -2.6470, -2.3725, -2.6554, -2.5950}};

  float dl[3][7] =
   {{-165.0710, -181.5454, -173.8222, -144.4215, -133.0235, -164.3357, -133.4750},
    {-164.3970, -167.8258, -165.6568, -166.2492, -150.3413, -165.3660, -164.6871},
    {-181.1707, -188.3021, -183.7980, -181.5388, -166.7875, -182.2059, -178.4917}};

  float el[3][7] =
   {{133.5685, 152.2116, 143.4666, 109.7186, 98.9686, 132.5978, 97.1853},
    {126.3626, 130.4036, 127.9483, 127.2383, 111.6787, 127.3099, 124.9140},
    {133.4351, 141.5502, 136.5114, 132.7192, 118.6037, 134.4543, 128.7903}};
  float emiss4 = 0.985;
  float emiss5 = 0.975;
/*------------------------------------------------------------------------*/

  if ((satellite != 7) && (satellite != 9) && (satellite != 11) &&
      (satellite != 12) && (satellite != 14) && (satellite != 15)  &&
      (satellite != 16)) {
    fprintf (stderr,
      "ERROR >>> SURFTEMP_CLEAR: UNSUPPORTED SATELLITE NUMBER, exiting\n");
    exit (-1);
  }

/* Convert satellite number to an index and use the channel 4 temperature
   to get an index for the correct temperature range.  */

  if (satellite == 7)
    sat = 0;
  else if (satellite == 9)
    sat = 1;
  else if (satellite == 11)
    sat = 2;
  else if (satellite == 12)
    sat = 3;
  else if (satellite == 14)
    sat = 4;
  else if (satellite == 15)
    sat = 5;
  else if (satellite == 16)
    sat = 6;

  if (tb4 < 240.)
    temprange = 0;
  else if ((tb4 >= 240.) && (tb4 < 260.))
    temprange = 1;
  else if (tb4 >= 260)
    temprange = 2;

  /* Operate on the appropriate surface type. */

   if (surftype == (char)LAND) {

       /* SNOW-FREE LAND */

       tsurf = al[temprange][sat] + bl[temprange][sat]*tb4 +
         cl[temprange][sat]*tb5 + dl[temprange][sat]*emiss4 +
         el[temprange][sat]*emiss5;
    }
    else {

       /* OCEAN, SNOW/ICE ALGORITHM */

       tsurf = a[hemis][temprange][sat] + b[hemis][temprange][sat] * tb4 +
         c[hemis][temprange][sat] * (tb4 - tb5)
         + d[hemis][temprange][sat] * (tb4 - tb5) *
         ((1.0 / cos (scanang * DEGRAD)) - 1.0);
    }

   return tsurf;
}
```