Clouds and the Earth's Radiant Energy System (CERES)

Data Management System

Software Design Document

Convolution of Imager Cloud Properties with CERES Footprint Point Spread Function (Subsystem 4.4)

ARCHITECTURAL DRAFT

Chris Currey[1] and
Chuck McKinley[2]

[1]NASA Langley Research Center
Mail Stop 423
Hampton, VA 23681-0001

[2]Science Applications International Corporation (SAIC)
One Enterprise Parkway, Suite 300
Hampton, Virginia 23666

Atmospheric Sciences Division
NASA Langley Research Center
Hampton, Virginia 23681-0001

Release 1.0
March 1996

Preface

The Clouds and the Earth's Radiant Energy System (CERES) Data Management System supports the data processing needs of the CERES science research to increase understanding of the Earth's climate and radiant environment. The CERES Data Management Team works with the CERES Science Team to develop the software necessary to support the science algorithms. This software, being developed to operate at the Langley Distributed Active Archive Center (DAAC), produces an extensive set of science data products.

The Data Management System consists of 12 subsystems; each subsystem represents a stand-alone executable program. Each subsystem executes when all of its required input data sets are available and produces one or more archival science products.

The documentation for each subsystem describes the software design at various stages of the development process and includes items such as Software Requirements Documents, Data Products Catalogs, Software Design Documents, Software Test Plans, and User's Guides.

This version of the Software Design Document records the architectural design of each Subsystem for Release 1 code development and testing of the CERES science algorithms. This is a PRELIMINARY document, intended for internal distribution only. Its primary purpose is to record what was done to accomplish Release 1 development and to be used as a reference for Release 2 development.

.

TABLE OF CONTENTS

LIST OF FIGURES

1.0 Introduction

The Clouds and the Earth's Radiant Energy System (CERES) is a key component of the Earth Observing System (EOS). The CERES instruments are improved models of the Earth Radiation Budget Experiment (ERBE) scanner instruments, which operated from 1984 through 1990 on the National Aeronautics and Space Administration's (NASA) Earth Radiation Budget Satellite (ERBS) and on the National Oceanic and Atmospheric Administration's (NOAA) operational weather satellites NOAA-9 and NOAA-10. The strategy of flying instruments on Sun-synchronous, polar orbiting satellites, such as NOAA-9 and NOAA-10, simultaneously with instruments on satellites that have precessing orbits in lower inclinations, such as ERBS, was successfully developed in ERBE to reduce time sampling errors. CERES will continue that strategy by flying instruments on the polar orbiting EOS platforms simultaneously with an instrument on the Tropical Rainfall Measuring Mission (TRMM) spacecraft, which has an orbital inclination of 35 degrees. In addition, to reduce the uncertainty in data interpretation and to improve the consistency between the cloud parameters and the radiation fields, CERES will include cloud imager data and other atmospheric parameters. The first CERES instrument is scheduled to be launched on the TRMM spacecraft in 1997. Additional CERES instruments will fly on the EOS-AM platforms, the first of which is scheduled for launch in 1998, and on the EOS-PM platforms, the first of which is scheduled for launch in 2000.

1.1 Document Overview

The CERES "Determine Cloud Properties, TOA and Surface Fluxes" Subsystem (Subsystem 4) has been divided into two separate subsystems. They are the "Determine Cloud Properties" Subsystem and the "Determine TOA and Surface Fluxes" Subsystem. The "Determine Cloud Properties" Subsystem has been further divided into two major components. Each component develops and publishes separate design documents, and develops separate sets of executable code. The first major component (Cloud Retrieval) combines the functions described in Algorithm Theoretical Basis Document (ATBD) 4.1, 4.2, and 4.3 (References 1, 2, and 3), and provides the cloud masks, cloud layers, and cloud properties for the cloud imager data on a per-pixel basis. The second major component is Convolution of Imager Cloud Properties with CERES Footprint Point Spread Function (PSF), corresponding to ATBD 4.4 (Reference 4). For convenience throughout the rest of this document, this component is referenced as "FOOTPRINT." FOOTPRINT accepts the pixel-level cloud data from the Cloud Retrieval component of the Determine Cloud Properties Subsystem and convolves it with the CERES footprint field-of-view (FOV) data obtained from either the Instrument Earth Scans (IES) or Single Satellite CERES Footprint TOA and Surface Fluxes (SSF) datasets. The "Determine TOA and Surface Fluxes" Subsystem (ATBDs 4.5 and 4.6) completes Subsystem 4.0 and the SSF archive product by producing fluxes at the top of the atmosphere (TOA) and at the surface (References 4 and 5).

The purpose of this document is to provide the design of the FOOTPRINT component of the Determine Cloud Properties Subsystem. The intended audience for this document is the CERES cloud subsystem teams, subsystem testers, neighboring subsystem teams, and science reviewers.

The objective of this document is to provide a complete set of Release 1 design specifications to guide the development of the FOOTPRINT component. The Release 1 design document addresses the requirements from the CERES Science Team's ATBD 4.4, as reflected in the CERES Data Management System Software Requirements Document (SRD) (Reference 7). The design document is developed by the CERES Data Management Cloud Subsystem Team. This document contains the following information:

- 1.0: an introduction
- 1.1: a document overview
- 1.2: a brief overview of the component
- 1.3: key concepts
- 1.4: implementation constraints
- 1.5: design approach
- 2.0: architectural design of the FOOTPRINT component
- references
- abbreviations, acronyms, and symbols

1.2 Subsystem Overview

The FOOTPRINT Subsystem's major objective is use high spectral and spatial resolution cloud imager data to determine cloud microphysical and optical properties within the larger CERES footprint. This provides a set of cloud properties optimally designed for studies of the role of clouds in the Earth's radiation budget and enables the cloud physical properties to be tied to the cloud broadband radiative properties in a consistent manner. This initial estimate of cloud properties is modified in Subsystem 5 to obtain consistency in cloud properties and TOA broadband radiative fluxes (Reference 8).

The FOOTPRINT Subsystem is divided into four major steps:

1. Read Process Control File data
2. Initialize data and files for FOOTPRINT execution
3. Process CERES Footprints
4. Write Output

The primary input data sets for the FOOTPRINT Subsystem are

1. The CERES Instrument Earth Scans data product contains time of observation, geolocation data, and filtered radiances for each footprint in spatial order. The CERES footprint effective diameter is 10 km for Tropical Rainfall Measuring Mission spacecraft and 20 km for EOS-AM and -PM spacecraft. The ERBE Instrument Validation Tape (IVT) data product is the test data set for Release 1 until an IES product is available. In a rerun condition, the SSF product is used in place of the IES, since all of the pertinent data from IES are carried forward into the SSF.

2. The cloud imager data from Advanced Very High Resolution Radiometer (AVHRR), Visible Infrared Scanner (VIRS), or Moderate-Resolution Imaging Spectroradiometer (MODIS) are processed by Subsystems 4.1-4.3 and passed to FOOTPRINT via the Pixel_Data file. This file represents a two-dimensional array (N scanlines by M pixels per scanline), with a data structure associated with each pixel containing pixel location, viewing geometry, observation time, multispectral radiance data, scene type, and cloud properties as determined in Subsystems 4.1 through 4.3.

The output science product is the intermediate SSF product. The intermediate SSF is subsequently processed and completed by Subsystems 4.5-4.6, and the resulting final SSF is an hourly CERES archival product that contains footprint geometry; radiance information; and the statistics for full footprint, clear footprint, cloudy footprint, and overlap footprint areas.

Detailed descriptions of IES and SSF are in the CERES Data Management System Data Products Catalog (DPC) (Reference 9).

The secondary output products are Visualization File (VISFILE), Status Message Facility (SMF) logs, and the Cloud Quality Control (CLDQC) report. VISFILE contains pixel-level data organized by CERES footprint for visualization and validation of the convolution process and algorithms. SMF logs contain system-level diagnostic messages related to the execution of the FOOTPRINT program and provide a mechanism for tracing run-time error conditions. The CLDQC report contains processing information, informative messages, and statistics.

CERES algorithms will be developed in three releases:

- Release 1 was operational and delivered to the Langley DAAC in March of 1996. This Release is designed to process global data from the existing ERBE/AVHRR/High Resolution Infrared Radiation Sounder (HIRS) data from the NOAA-9 and -10 spacecraft. This release is used to test algorithm concepts on global data and for comparing multiple algorithms for cloud parameters.

- Release 2 is scheduled to be ready by early 1997, in time for the TRMM launch of the first CERES instrument planned for August 1997 and will also be used for the EOS-AM1 launch in June 1998. Until the new CERES Angular Distribution Models (CADMs) are developed, the CERES analysis will rely on models based initially on the ERBE Angular Distribution Models (ADMs).

- Release 3 will utilize the new TRMM CADMs, which are expected to be developed during the first 18 months after the TRMM launch.

1.3 Key Concepts

The following key concepts are embodied in the FOOTPRINT Subsystem:

- Imager pixel
- CERES footprint
- Along-scan angle (δ)
- Cross-scan angle (β)
- Delta-offset ($\Delta\delta$)
- FOV centroid
- Data chunk
- Rolling data chunk
- Overlap data chunk
- Along-track angle
- Cloud height category
- Cloud overlap condition
- Point spread function

Imager pixel refers to a single cloud imager field-of-view, which ranges from 0.25 - 1 km for MODIS pixels, 2 km for VIRS pixels, and 4 km for AVHRR-Global Area Coverage (GAC) pixels.

CERES footprint refers to a single CERES field-of-view. Subsystem 4.4 convolves imager pixels into a CERES footprint and is a footprint-driven process. The size of a footprint varies with the viewing zenith angle of the scanner. At nadir with a Point Spread Function (PSF) half power cutoff, the CERES footprint size ranges from 9 km by 13 km on TRMM to 17 km by 27 km on EOS. At a 70-degree viewing zenith angle with a PSF 95 percent energy cutoff, the CERES footprint size ranges from 38 km by 116 km on TRMM to 71 km by 212 km on EOS.

Along-scan angle (δ) refers to the component of the angle, measured at the CERES instrument, between the vector to the FOV centroid and the vector to a specific imager pixel, parallel to the instantaneous motion vector of the CERES scanner. See Figure 1-1.

Cross-scan angle (β) refers to the component of the angle, measured at the CERES instrument, between the vector to the FOV centroid and the vector to a specific imager pixel, orthogonal to the instantaneous motion vector of the CERES scanner. See Figure 1-1.

Delta-offset ($\Delta\delta$) refers to the displacement along the along-scan axis between the axis of the optical FOV and the centroid of the footprint (FOV centroid), as shown in Figure 1-1.

FOV centroid refers to the location of the *median* value of the PSF distribution for the CERES FOV. For CERES this is located at the point where $\Delta\delta = -0.89$ degrees; for ERBE, the corresponding value is $\Delta\delta = -1.68$ degrees.

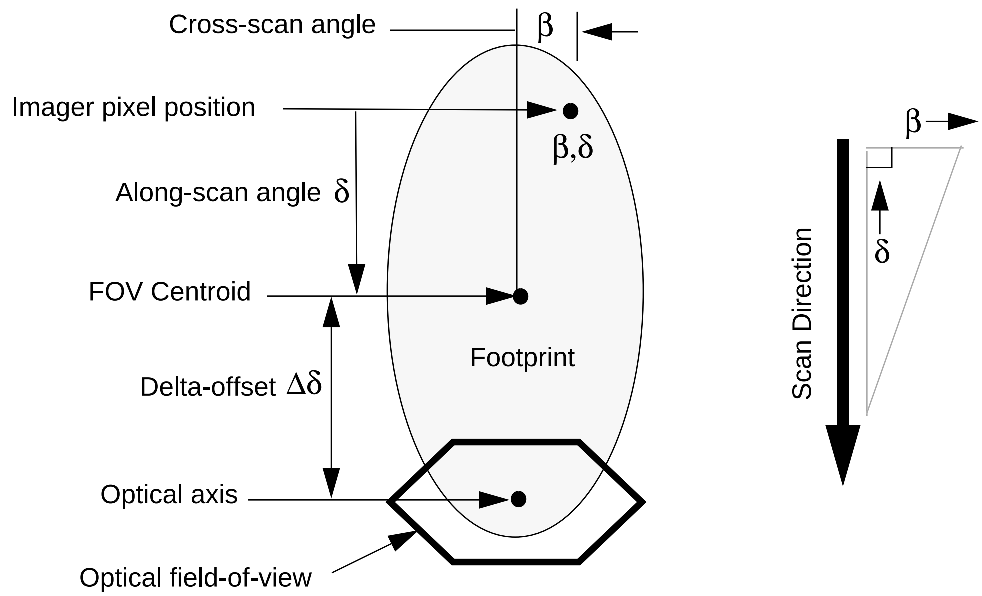

Figure 1-1. Scanner Footprint Geometry

The relationship between the imager pixels and the CERES footprint is conceptualized in Figure 1-2. The figure is exaggerated for clarity, but illustrates several features of the convolution process:

- The CERES footprints are defined by the envelope of the PSF which corresponds to a specified fraction of the total integrated PSF. This fraction is parameterized for testing and development, with an initial value of 95% of the total PSF energy. The corresponding PSF envelope value is 0.095, where PSF ranges from 0.0 to 1.0.

- There is considerable overlap between CERES footprints, so that any given imager pixel may be convolved into several succeeding CERES footprints in a given scanline, or into corresponding footprints from succeeding scanlines.

- The CERES footprints cover a wider crosstrack angle than the imager data swath, so that many of the footprints do not contain any imager pixels. Such footprints are not processed further by FOOTPRINT and do not appear in the SSF output product.

- In some cases, the CERES footprint are partially covered by imager pixels. FOOTPRINT has a parameterized secondary threshold to quantify the level of acceptability of partially covered footprints, in terms of the percentage of the total PSF energy represented by the imager pixels within the footprint. Partially covered footprints which exceed this threshold are processed by FOOTPRINT and flagged as partial. Footprints which do not meet the threshold are rejected and do not appear in the SSF.

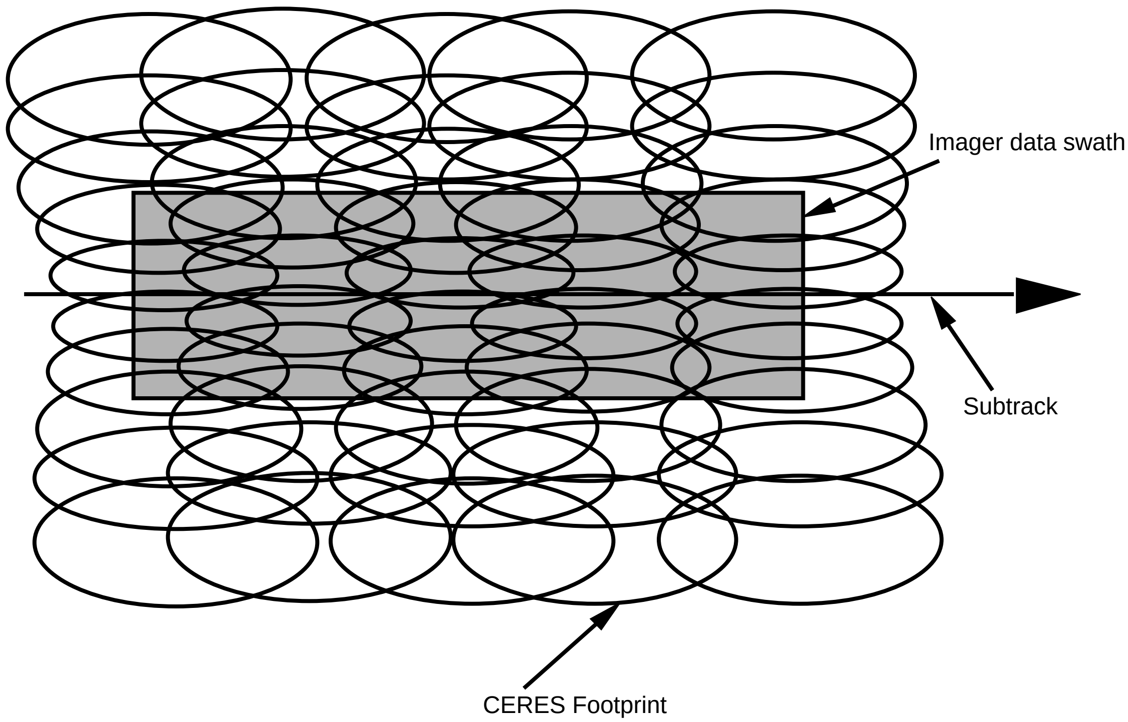

Figure 1-2. Relationship between CERES Footprints and Imager Pixels

The concepts of data chunk, rolling data chunk, and overlap data chunk are illustrated in Figure 1-3.

Several features of the FOOTPRINT data management scheme can be seen from Figure 1-3:

- *Data chunk* consists of a sequential series of scan lines of imager data, each scan line containing an equal number of pixels. FOOTPRINT requires a "*data chunk*" of imager data and its cloud properties for mapping image pixels onto the CERES footprint.

- CERES footprint records in the IES are arranged in spatial order, with monotonically increasing along-track angle of the field-of-view centroid with respect to the satellite subtrack. At the beginning and end of the hour, the CERES field-of-view may be at the nadir position (if in cross-track scan mode), or may be outside the imager data swath for the current hour (if in rotating azimuth plane mode). In either case, it is necessary to append imager data from the previous hour and the succeeding hour, sufficient to cover the maximum expected size and displacement of the footprints. This illustrates the *overlap* *data chunk* aspect of FOOTPRINT. Normally, the FOOTPRINT process does not start until all three consecutive Pixel_Data hourly input files are available for an IES hourly input file. However, FOOTPRINT can be run with minor data loss if either or both of the previous or succeeding hours of Pixel_Data are not available. This data loss varies with platform and scan mode, but is on the order of 1 percent of the total number of footprints available, if neither overlap hour is available.

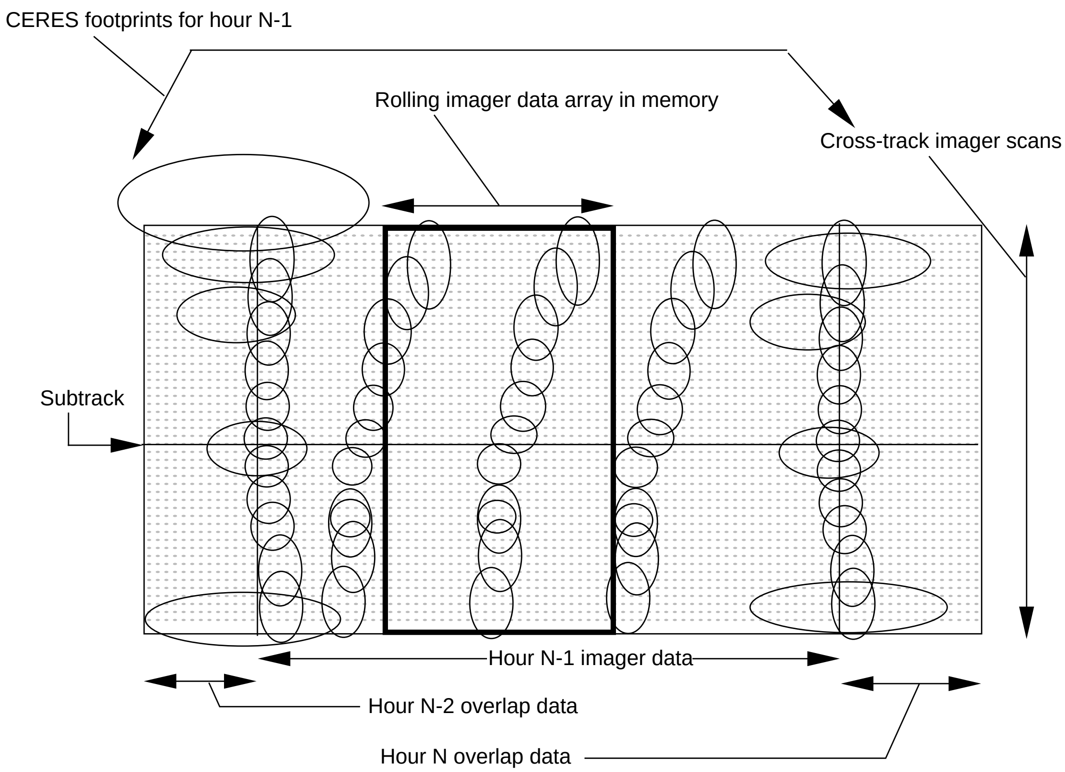

Figure 1-3. Data Chunk Concepts

- To minimize reading imager pixel data, FOOTPRINT holds in memory an array of pixel data large enough to contain the largest possible CERES footprint. Each time a new CERES footprint is read in, FOOTPRINT compares its along-track angle to the along-track angles of the first and last imager scanlines currently in memory. If additional data are needed at the forward end of the pixel data array, the necessary number of scanlines are added in the memory array, and an equal number are dropped from the back of the array, so that the array size remains constant throughout FOOTPRINT execution. This describes the *rolling data chunk* character of the pixel data array. Further, the progression is strictly in the forward direction due to the spatial ordering of the CERES footprints in the IES.

- FOOTPRINT processes the CERES footprints sequentially, one at a time, in the outer loop of a double loop. Processing of the imager pixels occupies the inner loop. In this way, any given imager pixel may be considered for convolution in several successive CERES footprints. Each CERES footprint, however, is processed only once, and is either included in the intermediate SSF as a fully convolved or partially convolved footprint or rejected due to insufficient coverage by the pixel data.

- Upon reading a CERES footprint record, FOOTPRINT first attempts to find an imager pixel within the PSF envelope surrounding the footprint field-of-view. If no pixel which meets the PSF threshold can be found in the memory array, the footprint is rejected. If a pixel is found which meets the PSF threshold, search logic is invoked to scan sequentially through the pixel data contiguous to the "first pixel," identifying all pixels which satisfy the PSF threshold requirements for convolution with the footprint. FOOTPRINT increments the statistical sums, counters, and histograms appropriate for the cloud properties and other characteristics associated with each pixel, and if the search logic determines that the footprint is adequately covered by the imager pixel data, the footprint record is completed and written to the intermediate SSF output. If not, the statistical accumulators are reset and a new CERES footprint is read in.

Cloud Category is one of four categories arranged vertically within the atmosphere according to effective pressure, p_e. The four cloud categories are defined as:

High	if $p_e <= 300$ hPa
Upper Middle	if 300 hPa $< p_e <=$ 500 hPa
Lower Middle	if 500 hPa $< p_e <=$ 700 hPa
Low	if $p_e > 700$ hPa

FOOTPRINT accumulates a variety of statistics of cloud properties for pixels with clouds in one or more of these categories and reports those statistics for each CERES footprint in the intermediate SSF output product. These statistics include such measures as PSF-weighted means and standard deviations, weighted area fraction for each cloud category, selected percentiles of optical depth or Infrared (IR) emissivity for each cloud category, and number of overcast imager pixels in each cloud category.

Cloud overlap conditions are the 11 combinations of 0, 1, or 2 cloud layers that a single pixel could represent and are defined as:

Clear
High
Upper Middle
Lower Middle
Low
High over Upper Middle
High over Lower Middle
High over Low
Upper Middle over Lower Middle
Upper Middle over Low
Lower Middle over Low

For each CERES footprint, FOOTPRINT calculates the fraction of pixels belonging to each of these overlap conditions and reports them in the SSF output.

Point Spread Function. PSF is a weighting function with values ranging from 1 at the centroid to 0 at points far off-axis. It is symmetrical in the cross-scan direction and slightly asymmetrical in the along-scan direction, as illustrated in Figure 1-4. A critical feature of the PSF is the offset between the peak PSF value and the optical centroid of the CERES field-of-view, which corresponds to the point where both cross-scan and along-scan angles are zero. This offset can be viewed intuitively to represent the lag time in the CERES instrument response function and is, therefore, specific to the CERES instrument. The CERES PSF is used as the primary selection criterion for inclusion of an imager pixel within a given CERES footprint by testing the PSF value for the imager pixel against a parameterized energy threshold value. In addition, each imager pixel is tested relative to a secondary parameterized PSF threshold to determine the acceptability of footprints which are only partially covered by imager pixels. For initial testing purposes, the energy levels for the primary and secondary thresholds are set at 95% and 75% respectively; the corresponding PSF envelope values are 0.095 and 0.42, respectively. During initial testing, both ERBE and CERES PSF distributions are used for comparative analysis and to test Subsystem results against historical ERBE data records.

1.4 Implementation Constraints

The design goal is to be flexible enough that subsequent releases can be addressed with minimum impact. Every attempt is made to design the Subsystem such that trading in and out of algorithms causes minimum change to the Subsystem organization. Therefore, the software design is modular and allows for optional processing choices. The processing options which are subject to the most frequent changes are controlled by fields in the Process Control File (PCF), permitting options to be selected at run time by editing the PCF, without requiring recompilation of the software.

The programming language used in FOOTPRINT is FORTRAN 90.

The design constraints are the EOS Data Information System (EOSDIS) Planning and Data Processing System (PDPS) operating environment, the Science Data Production (SDP) Toolkit, the limits of computer Central Processing Unit (CPU), data throughput, network capacity, and system complexity. Metadata and error handling will conform to the EOSDIS requirements and standards.

All bad data are expected to have been eliminated from the input data products (Pixel_Data and IES) or flagged as bad by the time FOOTPRINT is executed. Limit checks are implemented to ensure that the data are within reasonable limits. Data that are outside these limits are excluded from further processing, and a diagnostic message is generated.

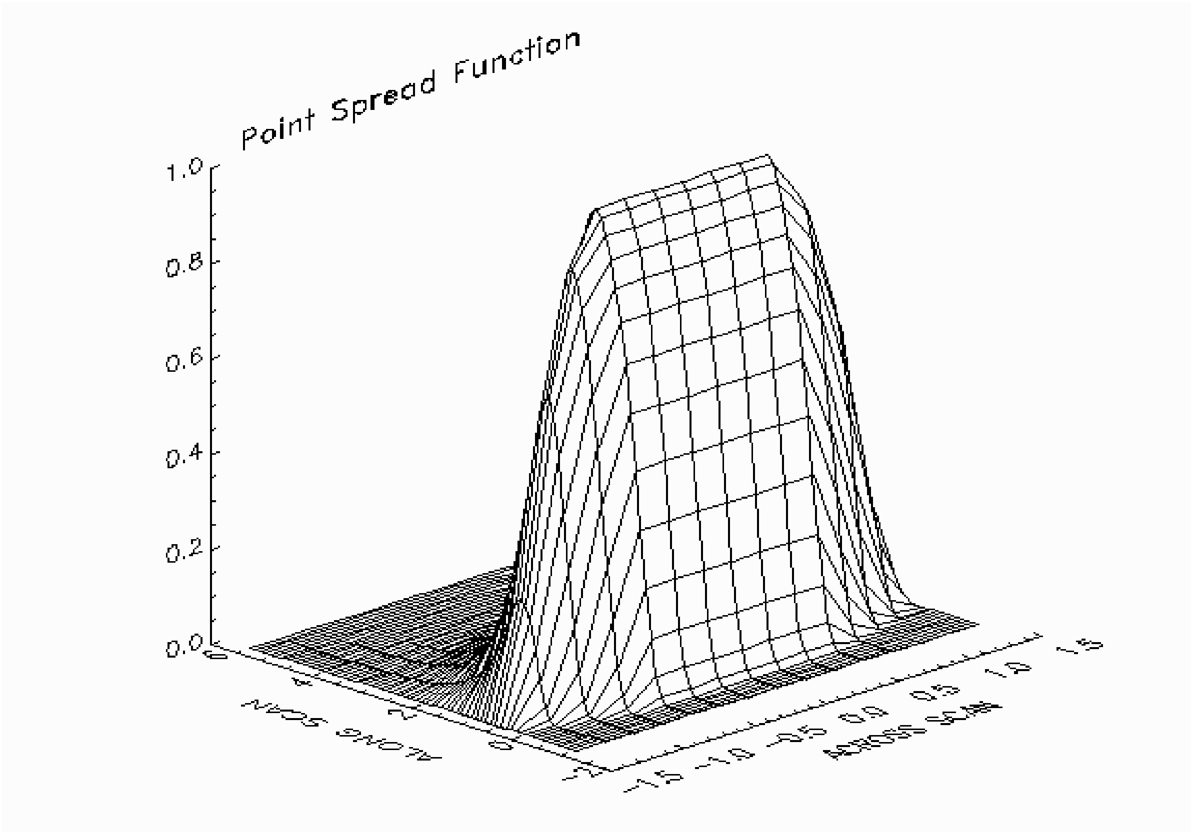

Figure 1-4. CERES Point Spread Function

A design issue is how to handle processing IES products containing a mixture of footprints obtained in the Rotating Azimuth Plane Scan (RAPS) mode and the Fixed Azimuth Plane Scan (FAPS) mode. In Release 2, FOOTPRINT will process these IES input products and produce two SSF products: one for the RAPS mode and one for the FAPS mode. Release 1 treats only the FAPS, or cross-track scan mode.

There are several design options to satisfy the Overlap Portions requirement that begins and ends the rolling data chunk. Normally, FOOTPRINT will not start until all three consecutive Pixel_Data hourly input files are available for an IES hourly input file. However, the initial design of FOOTPRINT supports execution for a given hour of Pixel_Data if either or both the preceding and succeeding hours of Pixel_Data are missing. A small data loss is incurred, corresponding to the area where the overlap data are not available. An option might be considered to save the data chunk from the previous hour into a small "overlap" file instead of searching through the previous hour file to a position ~200 km before the current Pixel_Data file. Saving an overlap file would change the requirement from three consecutive full-hour Pixel_Data files to two consecutive Pixel_Data hourly files in addition to the small overlap file.

1.5 Design Approach

Subsystem 4.4 generally follows the structured design methodology. However, due to intense interest in the feasibility and performance characteristics of certain key algorithms, the development of the design followed a vigorous incremental prototyping process. The following algorithms were individually prototyped and tested in early stages of the design evolution:

1. Point Spread Function - this function is the heart of the convolution process and must be calculated for each imager pixel location with respect to the centroid of each CERES footprint. The PSF algorithm is moderately compute-intensive so that speed of execution is essential. As an alternative, the use of a lookup table was tested. The results of these tests indicated that the lookup table was about four times faster than direct calculation using the explicit function. This function is highly modular in the sense that it can easily be replaced or modified as the CERES software matures.

2. Update_Pixel_Data_Array - this algorithm tests the along-track location of the current CERES footprint centroid against the limits of the imager data swath currently in memory. If the imager data swath does not contain enough data to cover the largest possible footprint centered at the along-track location of the centroid, it calculates the number of additional imager scanlines required to provide that coverage and updates the array with that number of scanlines. The array remains fixed in size, with the new scanlines overwriting the oldest scanlines in the array. Circular index logic is employed in addressing the scanlines.

3. Calc_FOV_Const - the purpose of this algorithm is to calculate the set of vector and scalar variables which are required as input for the PSF calculation and which remain constant for any one CERES FOV. The algorithm makes extensive use of vector analysis methods as described in ATBD 4.4 and the SRD (References 4 and 7).

4. Find_First_Pixel - the purpose of this algorithm is to locate, within the imager data swath currently in memory, any pixel ("first-pixel") which satisfies the requirement for convolution with the current CERES FOV. That requirement is simply that the PSF value for the pixel must exceed a specified level which defines the limit of the footprint.

5. Get_FOV_Pixels - the purpose of this algorithm is to perform a sequential search through the imager pixels in the current memory array, starting with "first-pixel," to find all pixels which satisfy the PSF threshold requirement for convolution with the current CERES FOV. This algorithm must be able to determine whether the footprint extends beyond the limits of the array. It must also recognize when the search is completed, e.g., when no more pixels in the array can satisfy the threshold requirements.

In addition to the algorithms noted above, interfaces for the required input and output files were developed at an early stage, and simulated input files were produced. The two critical input files for Subsystem 4.4 are the IES, which was simulated from ERBE IVT data, and the Pixel_Data (alias "CookieDough") imager data file, which was created by an early prototype version of Subsystem 4.1 using AVHRR GAC data. An interface for the intermediate SSF output file was also developed, and a utility was created to produce SSF files composed of a user-selectable number of artificial SSF records. These files were used by several subsystems to test SSF reading routines. The prototyping exercises were also important in establishing storage requirements associated with the input and output file sizes.

The programming language selected for development of Subsystem 4.4 is FORTRAN. In the initial prototyping work, FORTRAN 77 (f77) was used. However, with the subsequent availability of FORTRAN 90 (f90) compilers, the prototype software was converted to FORTRAN 90, and all later development was done with FORTRAN 90. The f90 language standard offers a number of advantages over FORTRAN 77. Two of the most significant advantages, used extensively in Subsystem 4.4 are

- FORTRAN 90 modules - these offer many specific advantages, including the elimination of common blocks, the minimization of calling arguments between routines, and the ability to organize software logically, with an object-oriented viewpoint;

- Declaration of data structures (FORTRAN 90 TYPES) including arrays of derived types. This was possible in FORTRAN 77 only by extensions to the language standard, which have been assimilated as part of the f90 standard. The data interfaces for both input and output data products in Subsystem 4.4 are greatly facilitated by use of data structures.

Initial prototype testing related to Subsystem 4.4 was conducted on a SUN Microsystems SPARC 2 platform using the SUN f77 compiler. With the implementation of the NAG 90 f90 compiler, all code was either converted or written directly in f90. A second stage of prototyping involved migration of the f90 code from the SUN to Silicon Graphics’ (SGI’s) Indy platform, again using the NAG 90 compiler. The final stages of prototype testing were performed on the SGI Challenge platform. Initially, SGI’s 64-bit f90 compiler was used for testing on this platform. However, subsequent tests using the NAG 90 32-bit compiler showed two significant advantages over the SGI compiler:

- Execution time for FOOTPRINT was cut approximately in half using the NAG compiler, as compared to the SGI compiler;

- The SGI compiler was troubled by a number of inconsistencies and apparent deviations from the f90 standard, forcing some code modifications and work-arounds.

Design efforts for Subsystem 4.4 were assisted by the use of the Software through Pictures (StP) software engineering tool, including the editors for Data Flow Diagrams, Structure Charts, and Data Structures, as well as the Data Dictionary and the Document Preparation System (DPS). This design document was produced using the Software Design Document (SDD) template of the DPS.

2.0 Architectural Design

The top-level Data Flow Diagram (DFD) for FOOTPRINT is presented in Figure 2-1. This DFD shows the three major processes in FOOTPRINT:

- The first major process requirement is to locate imager pixels which meet the criteria for convolution with a CERES footprint. CERES footprint data are read in sequentially from the IES, or in the case of a rerun, from a previously generated SSF file, which carries forward the same information contained in the IES. The imager pixel data are read into a fixed-size memory array, designated Pixel_Data in the DFD. The Pixel_Data files (termed "CookieDough" during initial development and testing) are produced by the previous Subsystems, 4.1-4.3, based on imager data taken from AVHRR, VIRS, or MODIS files. There is also an initiative to ingest Geostationary Operational Environment Satellite (GOES) data, but the current design would require some modification to accommodate that data.

- The second major process requirement is to calculate cloud property statistics for each CERES footprint, based on the cloud property information contained in the pixel data structure for each of the imager pixels which has been identified for convolution with the current CERES footprint. These statistics are written to the intermediate SSF output file in the form of CERES footprint-level data structures. Each record in the SSF output corresponds to one CERES footprint record in the IES (or SSF) input file. Note that many CERES footprints in the IES do not appear in the SSF as they fall outside the range of the imager data swath so that they are either devoid of imager pixel coverage or are insufficiently covered for meaningful statistics. If a run-time option has been selected to produce a validation product, this process also writes pixel-level data structures, organized by CERES footprints, to the auxiliary output visualization file. This file supports visualization and validation of the convolution process and algorithms by preserving the identity, location, and other selected attributes of all pixels convolved with each of the CERES footprints.

- The last process requirement is to produce a Cloud Quality Control report.

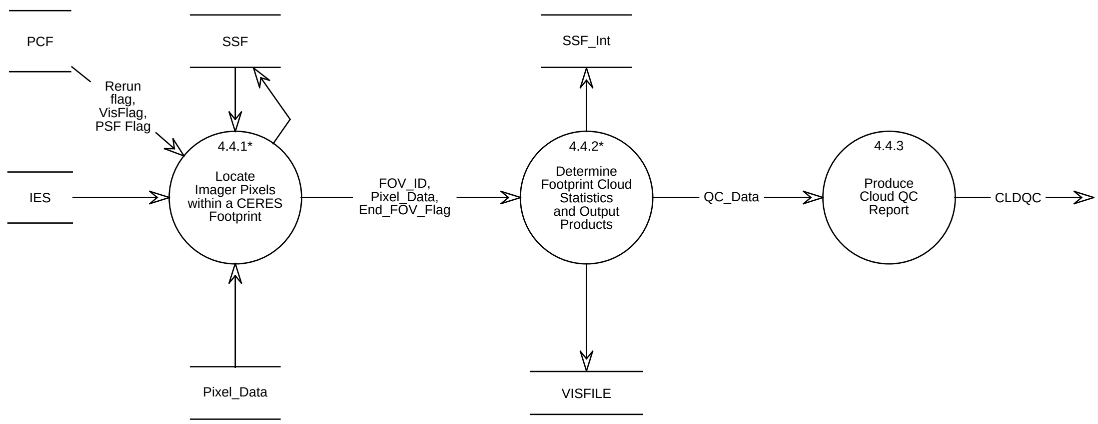

Figure 2-1. Top-level Data Flow Diagram

The structure of FOOTPRINT is relatively straightforward, as shown in the top-level structure chart in Figure 2-2.

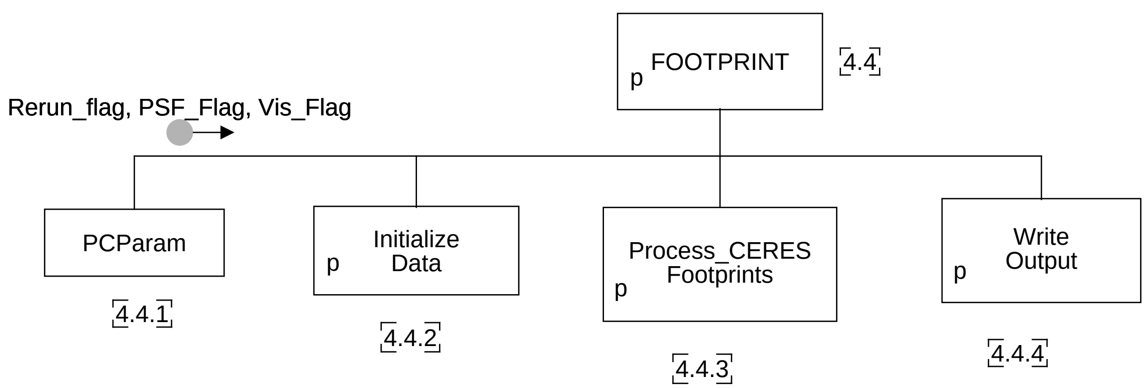

Figure 2-2. Top-level FOOTPRINT Structure Chart

- FOOTPRINT is the main driver routine for the program and simply calls each of the subordinate routines in sequence. In a normal execution, each of the subordinate routines shown in Figure 2-2 is called once and only once.

- PCParam opens the Process Control File and reads the run-time parameters contained therein. The PCF data identifies the data hour to be processed, the input filename, a flag specifying a normal run or rerun, a flag to indicate whether a pixel-level visualization file is to be written, and a flag to specify the use of either the ERBE or CERES PSF distribution.

- Initialize_Data opens all required input files, reads the header record in the IES or SSF input file to determine the platform data (satellite and instrument ID), and uses this information to set the values of a number of platform- and instrument-dependent global parameters. It creates and opens the intermediate SSF output file for direct access.

- The main processing module, Process_CERES_Footprints, whose decomposition is shown in Figure 2-3, is structured as a nested double loop. The outer loop is executed once for each CERES field-of-view record read sequentially from the IES (or SSF) input data file. The inner loop, contained in module Proc_Pixel_Data, is shown in Figure 2-4. It is executed once for each imager pixel record considered for convolution with the current CERES FOV. Logic within this module identifies whether there are any imager pixels within the CERES FOV. If not, the CERES FOV record is immediately rejected and the outer loop is executed for the next CERES FOV. If any imager pixel is found that satisfies the PSF threshold requirement for convolution with the CERES FOV, search logic within the inner loop is invoked to search sequentially through the pixels contiguous with the first pixel to identify and process those pixels which also satisfy the PSF threshold requirements. As each pixel is processed, the program increments the statistical accumulators pertinent to the data contained in the pixel data structure. These statistics are calculated on a CERES footprint basis. The search logic includes criteria to recognize when the search has been completed such that no other pixels in the current memory array will satisfy the PSF threshold requirements. At this point, the search is terminated, the statistics for the current CERES footprint are completed, and the footprint record is written to the intermediate SSF output file. Control is then returned to the Process_CERES_Footprints routine for the next execution of the outer loop. After the last footprint record in the IES (or SSF) input file has been processed, Process_CERES_Footprints terminates and control is returned to the FOOTPRINT driver routine.

- Write_Output writes a header record to index #1 in the intermediate SSF output file, including the number of footprint records written to the SSF file. Thus, there are N+1 records in an SSF file containing N footprint records. The header record is in record #1, and the footprint records are in records numbered 2 through N+1. Write_Output also writes the auxiliary output file CLDQC report.

One of the principal design features of FOOTPRINT is the extensive use of the FORTRAN 90 MODULE capabilities. The use of modules in FOOTPRINT serves largely to minimize the passage of calling arguments between routines, since shared data are defined in modules USEd (FORTRAN 90 "use" statement) by the routines requiring the data. This design uses no named or blank common blocks for passage of global data. As the Subsystem matures, it may be possible to further reduce the number of variables passed as calling arguments between routines.

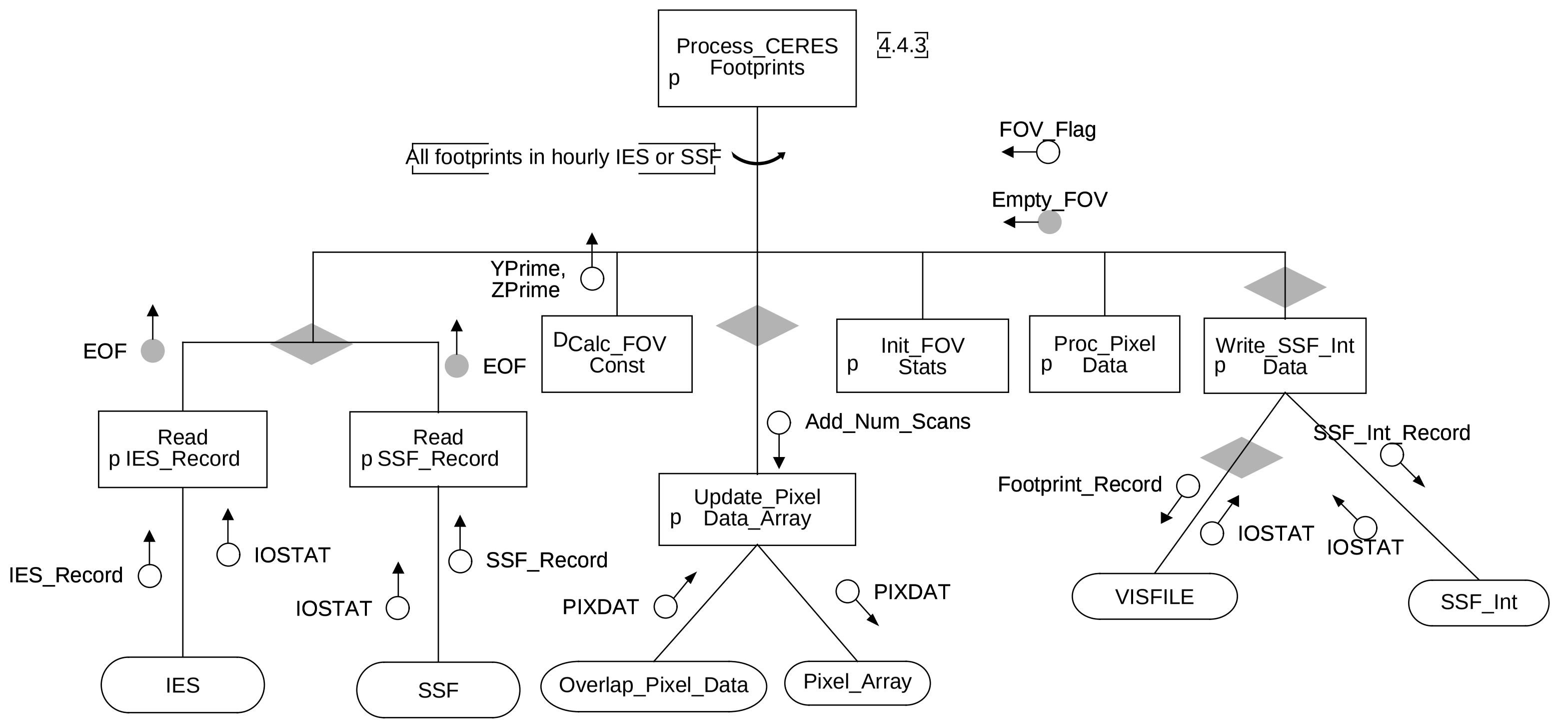

Figure 2-3. Process_CERES_Footprints Decomposition

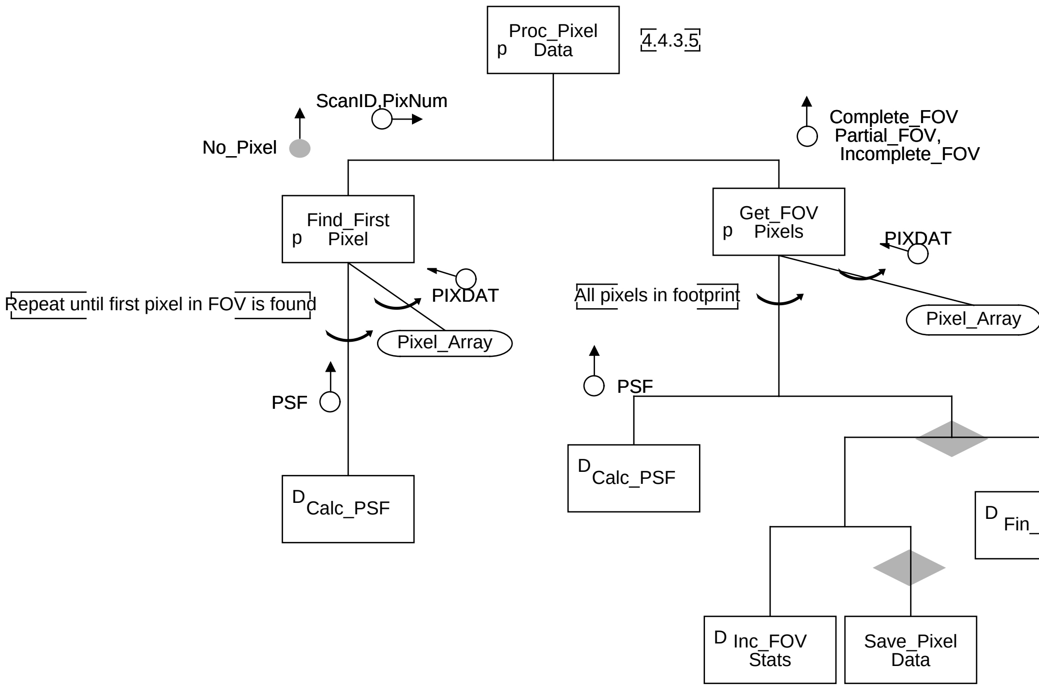

Figure 2-4. Proc_Pixel_Data Decomposition

The initial design takes advantage of the Process Control File to pass run-time parameters which may change on a run-by-run basis. The PCF is a human-readable (ascii text) file which can be edited by any text editor to configure a run. This permits runs to be made without recompilation of the source code and allows the use of UNIX scripts to execute sequences of runs automatically.

The structure chart decomposition for the Process_CERES_Footprints major loop in FOOTPRINT is presented in Figure 2-3.

Several of the routines shown in this figure deserve discussion:

- Process_CERES_Footprints is the driver for this branch of code. In addition, it compares the along-track angle of the current CERES FOV centroid against the along-track angles of the first and last nadir pixels in the pixel data array currently in memory. These angles are calculated from orbital parameters and timing data each time the pixel data array is updated. Process_CERES_Footprints converts the angular differences to linear measure and compares this to the parameterized maximum dimension of the CERES footprint. If this calculation indicates that there may not be enough pixel data at the forward end of the array to cover the footprint, the shortfall is calculated, converted to angular measure and thence to the equivalent number of scanlines of imager data, and subroutine Update_Pixel_Data_Array is called to roll that number of scanlines into the array. This test is performed for every new CERES FOV read in.

- Read_IES_Record/Read_SSF_Record sequentially read footprint records from either the IES input file or, in the case of a rerun scenario, the SSF file. The IES data structure is essentially a subset of the SSF data structure. These read routines map the data from either IES or SSF into a set of common internal variables defined in module CERES_TypDef.

- Calc_FOV_Const calculates a set of vector quantities related to the positions of the satellite and the current CERES FOV which remain constant throughout the processing of the current FOV, and which are needed for determination of the PSF value of each imager pixel considered for convolution with the FOV.

- Update_Pixel_Data_Array rolls new scanlines of imager data into the pixel data array in memory in response to calls from Process_CERES_Footprints. It copies the next ’N’ scanlines of data from the hour file of pixel data into the array, overwriting the ’N’ oldest array slices. Circular (modulus) arithmetic is used to keep track of the indices of the first and last scanlines of imager data in the array. In this way, it is not necessary to reindex the array every time an update is made.

- Init_FOV_Stats is an initialization routine, called for every new CERES FOV being processed. It resets all the statistical accumulators, counters, and histograms to zero in preparation for processing of the next FOV.

- Proc_Pixel_Data is the driver for the inner processing loop in the convolution process. Its decomposition is described following Figure 2-4.

- Write_SSF_Int_Data is the output branch for the Process_CERES_Footprint loop. It is called when Proc_Pixel_Data completes with an FOV_Flag indicating that the current CERES FOV satisfied the requirements for either full or acceptable partial coverage by imager pixel data. Write_SSF_Int_Data completes the assignment of internal data values to the fields of the SSF output data structure and writes the SSF record to the intermediate SSF output file. If the visualization flag (Vis_Flag) is TRUE, this routine completes the assignment of internal data values to the fields of the Footprint_Record data structure and writes the Footprint_Record to the visualization file.

The structure chart decomposition for the Proc_Pixel_Data inner loop in FOOTPRINT is presented in Figure 2-4.

Several of the routines shown in Figure 2-4 deserve discussion:

- Calc_PSF is at the heart of the convolution process, as the footprint is defined only by the imager pixels which satisfy the PSF threshold requirement to be included. Further, since PSF value is determined strictly by the along-scan and cross-scan view angles between the pixel location and the CERES FOV centroid, it must be calculated separately for each pixel and FOV centroid. Since there are typically 150 pixels (using AVHRR GAC data) in a CERES footprint and approximately 33,000 footprints in an hour of IES (using ERBE IVT data to create simulations), this routine is executed about 5 million times in processing an hour of data. In the postlaunch era, this number will increase in proportion to both the increased number of imager pixels in a given CERES footprint and the increased number of CERES footprints in an hour of data. As an idea of scale, for TRMM there will be about 150 million PSF calculations per hour of data, and for EOS-AM about 600 million. During initial testing, this routine was tested both as a direct calculation of the PSF value and as a table lookup. It was observed that the table lookup executed about four times as fast as the direct calculation, with negligible loss of accuracy. For all operational releases, Calc_PSF will, therefore, be configured as a table lookup.

- Find_First_Pixel attempts to find any pixel, within the pixel data array in memory, which satisfies the PSF threshold requirement to be included in the footprint. This routine uses a two-stage approach. In the first stage, the location of the current CERES FOV centroid and the geometry of the current pixel data array are used to directly calculate a scanline number and pixel index corresponding to a pixel whose colatitude and longitude are nearest to the FOV centroid; the PSF value for this pixel is calculated and compared to the threshold requirement; if it satisfies that requirement, the routine has successfully found a "first-pixel," and no further searching is required. However, if the PSF threshold requirement is not satisfied, the routine initiates an orderly search pattern, starting with the "first-pixel," and moving to adjacent pixels in all quadrants in such a way as to move progressively in the direction of decreasing along-scan and cross-scan view angles. This routine proved to be somewhat susceptible to infinite looping in prototype tests, so that a safeguard was incorporated to limit the number of search steps performed. Note that it is not permissible to search in the direction of increasing PSF value, for two reasons: 1) outside a narrow angular range around the FOV centroid, the PSF is defined everywhere to be 0.000, so that the search routine could not discriminate a direction of progress; 2) the PSF function has a plateau configuration in the cross-scan direction, with constant values over a relatively wide range of cross-scan view angles at any given value of the along-scan view angle. The search routine could, therefore, be caught in an infinite loop in the plateau region. In the course of prototype testing, it was observed that the first stage of this two-stage process tended to "miss" the FOV centroid location by about 15 pixels, very often not satisfying the PSF threshold requirements. In general, the second stage search was required, and several variations were tested. The variation finally selected was a compromise between searching efficiency and resistance to infinite looping. Testing also led to the addition of a logic branch to handle the case where the first stage calculates a scanline/pixel index combination outside the limits of the memory array, i.e., the FOV centroid is calculated to be outside the limits of the current pixel data swath; to avoid needlessly rejecting CERES footprints, the routine was modified to perform the stage two search, arbitrarily starting from the pixel at the midpoint of the memory array.

- Get_FOV_Pixels is the workhorse of the convolution process. Once the "first-pixel" has been found, Get_FOV_Pixels performs an ordered search, working outward from "first-pixel" in all directions to identify the pixels which satisfy the PSF requirement for inclusion in the footprint. This routine has logic to determine when the footprint overlaps the edge of the imager data swath; in such cases, the footprint may still be acceptable as a "partial" footprint, provided that all pixels satisfy a secondary threshold. If not, the footprint is rejected as incomplete and is not processed further. The routine also contains logic to determine when the search has been satisfactorily completed, i.e., when no more pixels within the pixel data array in memory can satisfy the PSF threshold requirements. At this point the routine terminates, and the final processing for the footprint is performed.

- Inc_FOV_Stats is the driver for the set of routines that accumulate the cloud property statistics for each CERES footprint. It is called from Get_FOV_Pixels each time a pixel is found that satisfies the PSF threshold requirements. Inc_FOV_Stats extracts the cloud properties and other data from the data structure associated with the pixel and increments the sums, sums-of-squares, counters, and histograms pertinent to the current CERES footprint record. It contains branches for processing of data pertaining to the full footprint area, the clear fraction, the cloudy fraction, and for eleven cloud overlap conditions.

- Fin_FOV is the driver for the set of routines that perform the final statistical processing when the last pixel in a footprint has been identified. Since all pixels are processed "on the fly," this set of routines calculates the means, standard deviations, percentiles, and other required statistics based on the running sums, sums-of-squares, counters, and histograms of the cloud properties.

- Save_Pixel_Data is a routine used only when validation data at the pixel level by footprint are desired. If selected (if the Vis_Flag in the Process Control file is set to 1), this routine writes a data structure, Footprint_Record, to the peripheral output file, VISFILE. This data structure contains location data for the CERES FOV as well as location data and selected science data for each pixel within the FOV. VISFILE is intended to be used for visualization and/or validation of the pixel data at the footprint level.

References

1. Clouds and the Earth’s Radiant Energy System (CERES), Algorithm Theoretical Basis Document, Imager Clear-Sky Determination and Cloud Detection (System Volume 4.1), Release 1.1, April 1994.

2. Clouds and the Earth’s Radiant Energy System (CERES), Algorithm Theoretical Basis Document, Summary of the Cloud Pressure Retrieval Methods (System Volume 4.2), Release 1.1, April 1994.

3. Clouds and the Earth’s Radiant Energy System (CERES), Algorithm Theoretical Basis Document, Cloud Optical Property Retrieval (System Volume 4.3), Release 1.1, April 1994.

4. Clouds and the Earth’s Radiant Energy System (CERES), Algorithm Theoretical Basis Document, Convolution of Imager Cloud Properties with CERES Footprint Point Spread Function (System Volume 4.4), Release 1.1, April 1994.

5. Clouds and the Earth’s Radiant Energy System (CERES), Algorithm Theoretical Basis Document, CERES Inversion to Instantaneous TOA Fluxes (System Volume 4.5), Release 1.1, April 1994.

6. Clouds and the Earth’s Radiant Energy System (CERES), Algorithm Theoretical Basis Document, Estimate Longwave and Shortwave Surface Radiation Budget (System Volume 4.6), Release 1.1, April 1994.

7. Clouds and the Earth’s Radiant Energy System (CERES), Software Requirements Document, Determine Cloud Properties (Subsystem 4.1 - 4.4), Release 1, Version 1, April 1995.

8. Clouds and the Earth’s Radiant Energy System (CERES), Algorithm Theoretical Basis Document, Compute Surface and Atmospheric Fluxes (System Volume 5.0), Release 1.1, April 1994.

9. CERES Data Management System Data Products Catalog. Release 1.0, August 1994.

Appendix A

Abbreviations, Acronyms, and Symbols

Appendix A
Abbreviations, Acronyms, and Symbols

ADM	Angular Distribution Model
ATBD	Algorithm Theoretical Basis Document
AVHRR	Advanced Very High Resolution Radiometer
CADM	CERES Angular Distribution Model
CERES	Clouds and the Earth's Radiant Energy System
CLDQC	Cloud Quality Control
CPU	Central Processing Unit
DAAC	Distributed Active Archive Center
DFD	Data Flow Diagram
DPC	Data Products Catalog
DPS	Document Preparation System
EOS	Earth Observing System
EOS-AM	EOS Morning Crossing Mission
EOSDIS	EOS Data Information System
EOS-PM	EOS Afternoon Crossing Mission
ERBE	Earth Radiation Budget Experiment
ERBS	Earth Radiation Budget Satellite
FAPS	Fixed Azimuth Plan Scan
FORTRAN	FORmula TRANslation
FOV	Field-of-view
GAC	Global Area Coverage (AVHRR data mode)
GOES	Geostationary Operational Environment Satellite
HIRS	High Resolution Infrared Radiation Sounder
IES	Instrument Earth Scans (CERES Internal Data Product)
IR	Infrared
IVT	Instrument Validation Tape
MODIS	Moderate-Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
PCF	Process Control File
PDPS	Planning and Data Processing System
PSF	Point Spread Function
RAPS	Rotating Azimuth Plane Scan
SDD	Software Design Document
SDP	Science Data Production
SMF	Status Message Facility
SRD	Software Requirements Document
SSF	Single Satellite CERES Footprint TOA and Surface Fluxes
StP	Software through Pictures
TOA	Top of the Atmosphere
TRMM	Tropical Rainfall Measuring Mission
VIRS	Visible Infrared Scanner
VISFILE	Visualization File