674-SO-43, Version 1.0,
SSC-PD-4028

SIRTF Science Center

Downlink Segment

Subsystem Design Specification

AOT Products Subsystem: SURSIMSLOPEI2

12 November 2000

California Institute of Technology
SIRTF Science Center

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

THIS PAGE IS INTENTIONALLY LEFT BLANK

SIRTF Science Center

Subsystem Design Specification

Prepared by:

Frank Masci

Concurred by:

Bill Latter Dave Shupe

Approved by:

Mehrdad Moshir

Concurred by:

Bill Green

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

Revision History

Version	Description	Date
1.0	Initial version	November 12, 2000

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

Table of Contents

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

List of Figures

List of Tables

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

1. Introduction

1.1. Purpose and Scope

The Subsystem Design Specification is a document that describes the basic requirements, assumptions, definitions, software-design details and necessary interfaces for each subsystem. The document will be used to trace the incremental development of each subsystem and also to allow trace-back of levied requirements; this document should have sufficient detail to allow future modification or maintenance of the software by developers other than the original developers. This document is an evolving document as changes may occur in the course of science instrument hardware design and maturity of operational procedures. This document is not intended to repeat sections or chapters from other Project documents; when appropriate, references to proper sections of primary reference documents will be made.

1.2. Document Organization

This document is organized along the major themes of Requirements; Assumptions; Operational Concept; Functional Descriptions; Functional Dependencies; Input; Output; Other S/S Interfaces; Algorithm Descriptions (when applicable); and Major Liens.

The material contained in this document represent the current understanding of the capabilities of the major SIRTF systems. Areas that require further analysis are noted by TBD (To Be Determined) or TBR (To Be Resolved). TBD indicates missing data that are not yet available. TBR indicates preliminary data that are not firmly established and are subject to change.

1.3. Relationship to Other Documents

The requirements on the operation of SIRTF flow down from the Science Requirements Document (674-SN-100) and the Facility Requirements Document (674-FE-100). The Science Operations System is governed by the SOS Requirements Document (674-SO-100). The current document is also cognizant of the requirements that appear in the Observatory Performance and Interface Control Document (674-SEIT-100) as well as the Flight Ground Interface Control Document (674-FE-101). This document is also affected by the FOS/SOS Interface Control Document (674-FE-102) that governs interfaces between the Flight Operations System and the Science Operations System. Related Software Interface Specifications (SIS) will be as indicated in Section 2.2 of this document.

1.4. Change Procedure

This document is a level 4 document according to the SIRTF Project Documentation Plan (674-FE-103). Changes to this document after approval require the approval of the SOS Change Board (TBD). The process for change control is described in the SOS Configuration Management Plan.

2. Overview

The SURSIMSLOPEI2 program reads a FITS cube composed of "sample-up-the ramp" data represented by a set of non-destructive reads and converts this to a two plane FITS cube. The first output plane is a slope image derived using a linear least squares algorithm and the second plane is a difference image of the first two planes in the input cube. The input cube is often referred to as "RAW-mode" acquisation data and the output produced from this program, the "SUR-mode" image data.

Both planes in the output cube have pixel values in units of data number (DN) per sampling-time interval in signed 16-bit (I*2) integer format. The sampling time is the time separating each consequetive plane in the input (RAW-mode) FITS cube. This program is designed to simulate the SUR-mode data taking mode (downlinked in I*2 format), however, a more general program was also written which computes a SUR-mode image cube in R*4 (32-bit) format (called SURSIMSLOPER4). The purpose of this latter program is for collapsing RAW-mode cubes that have undergone pipeline processing. SURSIMSLOPEI2 is written in standard C.

2.1. SURSIMSLOPEI2 Requirements

SURSIMSLOPEI2 is initiated by a startup script under the control of the pipeline executive and does its required functions for a given DCE image or pre-processed DCE image; this involves performing the following tasks.

A.) Retrieve the command line parameters passed by the start up script and use them to run the program.

B.) Read in as input a standard RAW-mode FITS cube.

C.) Produce as primary output a 2-plane SUR-mode FITS cube containing simulated slope-image and difference-image planes.

D.) Provide exit codes to the pipeline executive and also provides logon and logoff messages identifying the version number and write any error messages to the standard output devices.

E.) Produce a processing summary.

2.2. Applicable Documents

The following documents are relevant to the SURSIMSLOPEI2 program of the AOT PRODUCTS Subsystems.

A.) The SOS Requirements Document

B.) The SOS Downlink Requirements Document

C.) The SOS Downlink Software Development Guidelines

D.) The following Software Interface Specifications (SIS)

SFO-SIS-3010 (16-bit integer RAW-mode DCE data)

TBR…

2.3. Version History

2.3.1. Version 1.0

Initial version created on November 12, 2000.

2.4. Liens

No liens have been identified

3. Input

3.1. SURSIMSLOPEI2 Input

SURSIMSLOPEI2 takes all of its input from either the command line or namelist file, which is set up by the startup script that is controlled by the pipeline executive or standalone. If the namelist is not specified, then all required inputs are expected from the command line. If both namelist and command-line inputs are specified, then the command-line inputs override the namelist values. Prior to reading namelist and/or command-line parameters, default values for the relevant parameters are assigned.

3.1.1. SURSIMSLOPEI2 NAMELIST Input

SURSIMSLOPEI2 reads the NAMELIST file whose name is passed to it by start-up script. The name of the NAMELIST is SURSIMSLOPEI2IN. The parameters that can be defined in the NAMELIST are listed in Table 1.

Namelist variable	Description	Dim.	Type	Units	Default
FITS_Image_Filename	Input n-plane FITS-image cube	161	C	-	Null
FITS_Out_Filename	Output 2-plane FITS-image cube.	161	C	DN per Δt	Null
Log_Filename	Output log filename	161	C	-	Stdout
Ancillary_File_Path	Pathname where supporting source files are installed.	161	C	-	./ (current directory)
Start_Data_Plane	First data plane to process in input FITS-image cube.	1	I*4	-	1

Table 1. Namelist file

The following is an example of the contents of a SURSIMSLOPEI2IN NAMELIST file that might be used, where the values specified are not necessarily realistic.

```
&SURSIMSLOPEI2IN
 Comment = 'Generic namelist file for sursimslopeI2, default values.',
 Ancillary_File_Path = '../sursimslopeI2_v1',
 FITS_Image_Filename = '../../test_imgs/MIPS_24_RAWstimtest.fits',
 FITS_Out_Filename = './testing/SUR_mode.fits',
 Log_Filename = 'stdout',
 Comment = 'First data plane in RAW cube to commence processing:',
 Start_Data_Plane = 3,
&END
```

3.1.2. SURSIMSLOPEI2 Command-Line Input

Alternatively, all inputs can be specified via command line, in which case, a namelist file is not needed. Or, inputs can be provided with a hybrid of both namelist and command-line mechanisms, with the latter overriding the former. Table 2 lists the available command-line options associated with their namelist-variable counterparts, as well as other options for specifying the namelist-file name and making the standard output more verbose.

3.1.3. SURSIMSLOPEI2 FITS Input

SURSIMSLOPEI2 uses the FITSIO library routines to read in the FITS-formatted input data file. The routines used are: fits_open_file, fits_read_keys_lng, fits_read_keys_dbl, fits_read_img, and fits_close_file.

Command-line option	**Value**
-n	Namelist_Filename
-i	FITS_Image_Filename
-o	FITS_Out_Filename
-l	Log_Filename
-a	Ancillary_File_Path
-p	Start_Data_Plane
-v (verbose switch)	-
-vv (super-verbose switch)	-

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

-d (debug switch)	-

Table 2. Command-line options

4. Processing

4.2 SURSIMSLOPEI2 Processing

SURSIMSLOPEI2 begins processing by writing its name and version number to standard output (verbose mode only), and then it initializes relevant variables with defaults values, and checks that the required namelist parameters and/or command-line parameters were passed to it. If this condition is not true, then it writes a message stating which parameters are missing, recommends a look at this document, and terminates by issuing an appropriate exit code to the pipeline executive; otherwise it proceeds as follows.

If an error occurs during processing, then an error message is written to standard output, a termination-status code is written to the log file, and an exit code to the pipeline executive issued.

After processing, the program name and version number, namelist filename (if used), input, and output filenames, values of other input parameters, date and time, processing time, and a termination-status code are written a log file.

4.3 SURSIMSLOPEI2 Processing Phases

SURSIMSLOPEI2 operates in six phases: initialization, data input, linear least square slope computation, difference image computation, results output, and termination. This processing level is depicted in Figure 1.

4.3.1. SURSIMSLOPEI2 Initialization

SURSIMSLOPEI2 initializes itself by performing the following tasks.

A.) A message is printed to STDOUT (verbose mode only), which includes the program name and version number.

B.) If specified on the command line, the NAMELIST file is opened and read. If any errors are encountered, a message is printed, and execution aborts.

C.) The remaining command-line inputs are read and checked for correct data range, consistency, etc. If any errors are encountered, a message is printed, and execution aborts.

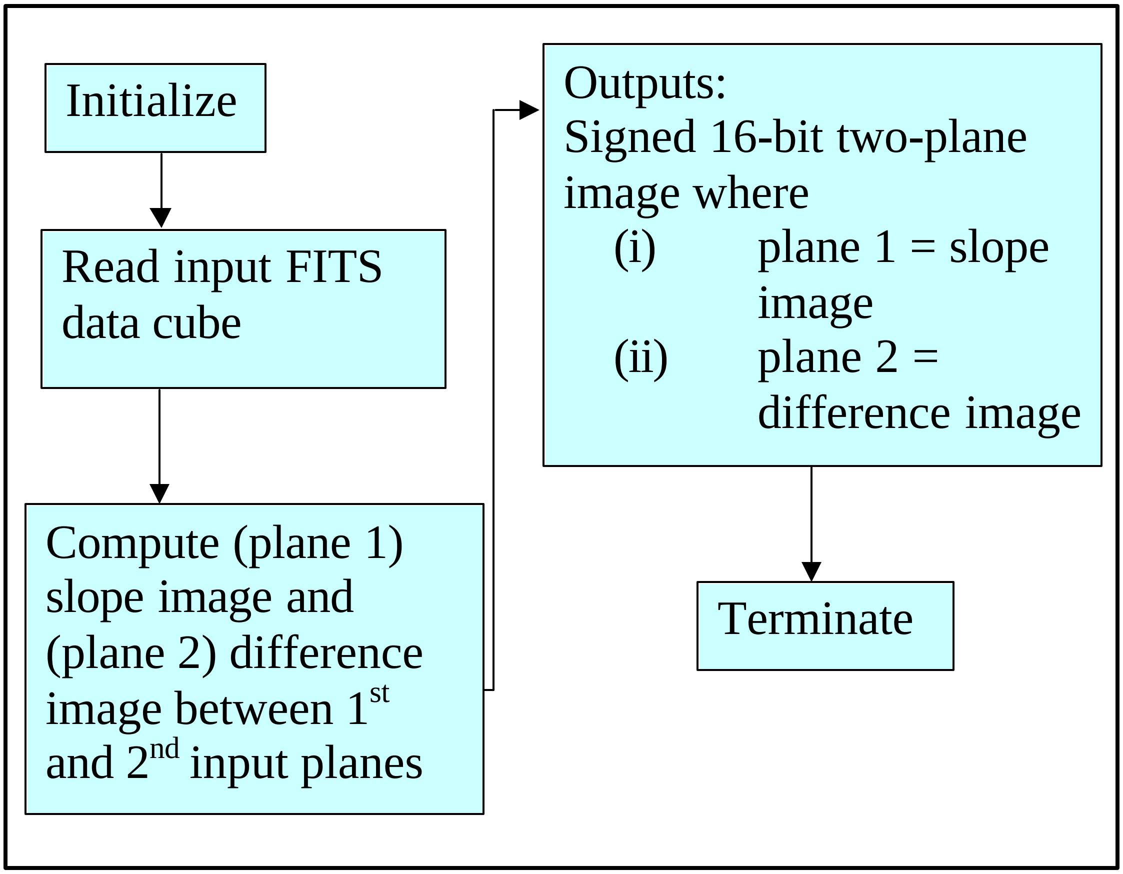
Initialize
Read input FITS data cube
Compute (plane 1) slope image and (plane 2) difference image between 1st and 2nd input planes
Outputs:
Signed 16-bit two-plane image where
(i) plane 1 = slope image
(ii) plane 2 = difference image
Terminate

Figure 1. SURSIMSLOPEI2 data and processing flow

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

4.3.2 FITS-Image Input

The input image is read and stored in memory. This includes all data planes in the input FITS cube, however, depending on the –p option setting, processing will commence at a user-specified plane number in the input FITS cube.

4.3.3. Slope-Image Plane Computation

The DN value for the same pixel in each data plane together with the corresponding "ramp-up" time t_k comprise one independent data set from which to derive a slope fit using a linear least squares method. The "ramp-up" time is given by:

$$t_k = k\ T\_INT,$$

where T_INT is the sampling integration time encoded in the input FITS cube header and k is the input plane number. Plane number 1 is defined with $k=1$. Given DN as a function of t, the problem consists of fitting a straight line model:

$$DN(t) = s\ t + y_{int}$$

to a set of N data points: t_k , DN_k , where s is the slope and y_{int} the intercept. We are primarily interested in the slope, which, using the general linear least square method is estimated by:

$$s = \left[\Sigma_k (t_k - (1/N)\ \Sigma_k t_k) DN_k\right] / \ \sigma_N$$

$$s_N = \Sigma_k (t_k - (1/N)\ \Sigma_k t_k)^2$$

(see Numerical Recipes in C, Press et al. 1988 for a complete derivation).

The slope is computed for every pixel with data from all planes in the input cube to create a "slope image". However, the user can also specify which plane to process first via the Start_Data_Plane parameter. All plane numbers prior to this value will not be included in computation of the slope image. Pixel values in the slope plane image are in units of DN per sampling time interval. The sampling time interval is given by the T_INT header keyword in the input FITS data cube.

4.3.4. Difference Image Plane Computation

The difference image plane is computed by subtracting plane number Start_Data_Plane from plane number (Start_Data_Plane + 1) in the input FITS cube. Start_Data_Plane is a user-specified parameter (see above). The DN pixel values are then re-scaled in terms of DN per sampling time interval.

4.3.5. FITS-Image Output

Processing statistics are given in both the standard output and log file. The calculation results are given in a 16-bit/pixel two-plane FITS image file.

4.3.6. Termination

Summary output is appended to the log file (the log file is created if previously non-existent), which includes diagnostic reports for the Q/A Subsystem and the appropriate exit code issued to be picked up by pipeline executive. A detailed list of log file contents is given in Section 6.1.3.

5. Algorithm

5.1. Algorithm Description

The simple algorithm employed in this software has been adequately described in the previous section. As a detail, pixels with NaN values in the input planes to be processed are "preserved" and carried through to the output image planes. NaN values are skipped and not included in computation of the slope-image and difference-image for that pixel. The number of NaN pixels in the output image is written to stdout.

5.2 Algorithm-Implementation Details

If pixels with values at the 16-bit saturation limits [-32768, 32767] are encountered in any of the input planes, a warning can be printed to stdout indicating the pixel location and its corresponding plane number. This warning is only produced if the “super-verbose” command-line (-vv) option is set (see Table 2). Processing then continues as normal with inclusion of the saturated pixel.

Furthermore, if the computed pixel values in either the slope or difference image planes exceed the allowable (signed) 16-bit range [-32768, 32767], a warning is written to stdout indicating the pixel location, the value is forced to the saturation value 32767 and processing continues as normal. This warning is automatically produced regardless of any “verbose” command-line option settings on execution.

6. Output

6.1. SURSIMSLOPEI2 Output

SURSIMSLOPEI2 is capable of generating the following output:

A.) Standard-output processing and status messages.

B.) A 16-bit integer two-plane FITS image representation of pixels in terms of slopes (plane 1) and a difference between the first and second input planes (plane 2). Pixel values are in units of DN per sampling time interval (T_INT) in both output planes.

C.) A log file containing processing statistics and status messages.

All SURSIMSLOPEI2 disk output is written to the pathnames that are specified with the output filenames in the command-line or namelist inputs.

6.1.1 SURSIMSLOPEI2 FITS Output

SURSIMSLOPEI2 uses the FITSIO library routines to create FITS-formatted output data files. The routines used are: fits_read_key_lng, fits_insert_key_lng, fits_create_file, fits_open_file, fits_copy_hdu, fits_flush_file, fits_write_key, fits_update_key, fits_write_date, fits_write_key_str, fits_write_key_fixflt, fits_write_img, fits_get_hdrspace, fits_read_record, fits_write_record, and fits_close_file.

6.1.2 SURSIMSLOPEI2 Log-File Output

The information stored in the log file at the output of this program includes: program name and version number, values of all namelist and/or command-line inputs, a message indicating the type of calculation performed, status code, processing time, date and time, and a message indicating program termination.

7. Testing

SURSIMSLOPEI2 has been successfully unit-tested as a stand-alone program for a variety of different input cases. The tests were designed to check SURSIMSLOPEI2 robustness and capability of generating corrected results.

Here is a summary of the unit tests that were conducted:

1. Executed SURSIMSLOPEI2 with inputs read from and output written to directories different from where the program was run. Both namelist and command-line input mechanisms were exercised.

2. Executed SURSIMSLOPEI2 with input image cubes consisting of MIPS (RAW-mode) test data with and without saturated pixels.

3. Executed SURSIMSLOPEI2 for all combinations of input parameters, in order to test that they function properly.

8. Usage Examples

Using a namelist file with verbose (-v) output re-directed to a file “out.log”:

```
SURSIMSLOPEI2 –n sursimslopeI2.nl -v | & tee out.log
```

Without using a namelist file:

THIS IS A PRELIMINARY DOCUMENT, the module described here may or may not be utilized in the final pipelines as described.

```
SURSIMSLOPEI2 -i input.fits -v -a ../ancpath –p 3 –o output.fits
```

9. Glossary

DCE	Data Collection Event
DN	Data Number
IOC	In-Orbit Checkout
SDS	Subsystem Design Specification
SIS	Software Interface Specification
TBD	To Be Determined
TBR	To Be Resolved